

AperTO - Archivio Istituzionale Open Access dell'Università di Torino

Verification and Control for Probabilistic Hybrid Automata with Finite Bisimulations

This is the author's manuscript

Original Citation:

Availability:

This version is available <http://hdl.handle.net/2318/1681478> since 2019-11-22T09:03:22Z

Published version:

DOI:10.1016/j.jlamp.2018.11.001

Terms of use:

Open Access

Anyone can freely access the full text of works made available as "Open Access". Works made available under a Creative Commons license can be used according to the terms and conditions of said license. Use of all other works requires consent of the right holder (author or publisher) if not exempted from copyright protection by the applicable law.

(Article begins on next page)

Verification and Control for Probabilistic Hybrid Automata with Finite Bisimulations

Jeremy Sproston

Dipartimento di Informatica, Università degli Studi di Torino, Corso Svizzera 185,
10149 Torino, Italy


Abstract

A hybrid automaton is a formal model for a system characterised by a combination of discrete and continuous components. Probabilistic hybrid automata generalise hybrid automata with the possibility of representing random behaviour of the discrete components of the system, in addition to nondeterministic choice regarding aspects such as time durations between mode switches and gradients of continuous flows. Two standard problems for probabilistic hybrid automata are verification and control: verification concerns the existence of a resolution of nondeterminism such that the resulting probability of an ω -regular property exceeds some bound; control concerns the existence of a resolution of the controllable nondeterminism, however the uncontrollable nondeterminism of the environment of the system is resolved, such that the probability of an ω -regular property exceeds some bound. While simple verification and control problems for (probabilistic) hybrid automata are in general undecidable, previous work has defined various subclasses for which the problems are decidable. In this paper, we generalise previous results by showing how bisimulation-based finite abstractions of non-probabilistic hybrid automata can be lifted to the setting of probabilistic hybrid automata. We apply these results to the subclass of probabilistic rectangular hybrid automata in a semantics in which discrete control transitions can occur only at integer points in time. These results allow us to show that, for this class of probabilistic hybrid automaton, the verification problems and control problems are decidable.

1 Introduction

Systems that are characterised by the interplay between discrete and continuous components are called hybrid systems. An example of a hybrid system is that of a digital controller embedded in an analog environment; this kind of system can be found in a wide variety of contexts, such as manufacturing processes, automotive or aeronautic applications, and domestic appliances. The critical nature of such systems, both from a social and an economic viewpoint, has led to the development of formal techniques to support the systems' correct construction. For this purpose, formalisms for hybrid systems, such as hybrid automata [ACH⁺95], have been introduced, along with associated analysis techniques. A hybrid automaton consists of a finite control graph, to model the discrete components, equipped with a finite set of real-valued variables, to model the continuous components. The graph is annotated with constraints on variables in order to describe the interaction of the discrete and continuous components. As time passes while control remains within a node of the graph, the values of the variables change continuously according to differential equations associated with the node. At certain points in time, control can instantaneously jump from one node to another, and the variables either retain their current value or change discontinuously with the jump. Automatic analysis techniques for hybrid automata generally belong to two categories: *verification* approaches, such as those based on model checking (see, for example, [ACH⁺95, Hen96]), consist of determining whether the hybrid automaton satisfies some correctness property; *controller-synthesis* approaches involve the computation of a control strategy for (some of) the digital components of the system such that the application of this strategy guides the

Published in *Journal of Logical and Algebraic Methods in Programming*, doi:10.1016/j.jlamp.2018.11.001

© 2018. This manuscript version is made available under the CC-BY-NC-ND 4.0 license <http://creativecommons.org/licenses/by-nc-nd/4.0/> 

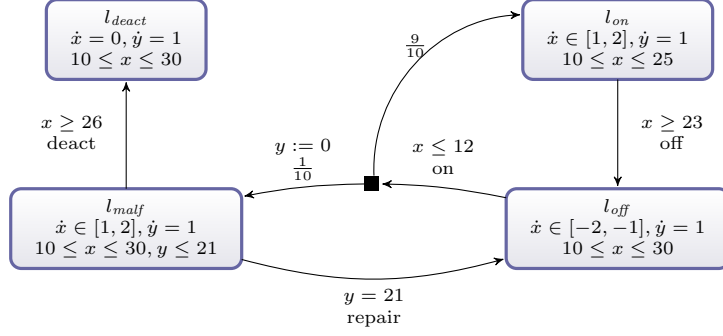


Figure 1: A probabilistic hybrid automaton modelling a faulty thermostat

system in order to guarantee the satisfaction of some correctness property, no matter how the environment behaves (see, for example, [WT97, HK99, HHM99, ABD⁺00]).

The basic hybrid automaton formalism does not take into account the relative likelihood of system events. Consider that, for example, in a manufacturing process a physical component may break, or in an aeronautic application there may be exceptional weather conditions, in both cases with low probability. We may wish to represent such events within our hybrid automaton model, together with the information about their probability of occurrence. This has led to interest in probabilistic extensions of hybrid automata, where probabilistic information is added in a number of different ways [HLS00, Spr00, Spr01, Buj04, APLS08, KR08, AD10, FHH⁺11, HNP⁺11, ZSR⁺12, LAB15, JR16]. In this paper, we consider *probabilistic hybrid automata*, as considered in [Spr00, Spr01, HNP⁺11, ZSR⁺12], which extend hybrid automata with probabilistic choices over the discrete part of the system. This formalism permits the modeling of events such as faults and message losses, in addition to randomized choices made by the digital components.

Example 1. An example of a probabilistic hybrid automaton modelling a faulty thermostat is shown in Figure 1. We use a number of the usual conventions for illustrating hybrid automata and probabilistic systems, such as \dot{x} to refer to the first derivative of variable x , and a black square to denote a (non-trivial) probabilistic choice. Nodes of the graph are referred to as locations. The ambient temperature is represented by the variable x , and variable y is a timer. When the heater is on (location l_{on} or location l_{malf}), the temperature increases at a rate between 1 and 2; when the heater is off (location l_{off}), the temperature changes at a rate between -2 and -1 . The locations l_{on} and l_{off} correspond to non-faulty behaviour, whereas the location l_{malf} corresponds to the heater being on in the presence of a fault in the temperature sensor that means that the measurement of the temperature is temporarily unavailable. The system passes from l_{on} to l_{off} , with probability 1, when the temperature is between 23 and 25, and from l_{off} to l_{on} , with probability $\frac{9}{10}$, or to l_{malf} , with probability $\frac{1}{10}$, when the temperature is between 10 and 12. The sensor fault means that the temperature can increase to a higher level in l_{malf} than in l_{on} . After a malfunction, either the system is deactivated if the temperature reaches 26, i.e., an excessive level (location l_{deact}), or the system times-out exactly 21 time units after the location l_{malf} was entered, in which case the heater is switched off. All edges of the probabilistic hybrid automaton correspond to making a transition to a certain location with probability 1, apart from the probabilistically branching edge from l_{off} .

In this paper we consider exact abstraction methods for probabilistic hybrid automata, which generally consider the construction of a finite-state probabilistic system (more precisely, a probabilistic automaton [Seg95] or a Markov decision process [Put94]) that represents faithfully the behaviour of the original system. Our approach is to provide a common framework for previous results for restricted subclasses of probabilistic hybrid automata, such as probabilistic timed automata [GJ95, KNSS02] and probabilistic multisingular and α -minimal automata [Spr00], using probabilistic bisimulation [LS91, SL95]. Probabilistic bisimulation is an equivalence relation that, for certain classes of probabilistic hybrid automata, can be used to obtain a finite number of equivalence classes, each containing a potentially infinite number of states, from which an equivalent finite-state system can be constructed and analysed using standard techniques for finite-state probabilistic systems. Our approach is based on the following key

property: any probabilistic hybrid automaton can be translated into a non-probabilistic hybrid automaton in which information concerning probability distributions is encoded in labels on edges of the graph. Consider a probabilistic hybrid automaton \mathcal{H} : we show that if the *non-probabilistic* hybrid automaton counterpart of \mathcal{H} has a finite *non-probabilistic* bisimulation equivalence quotient, then \mathcal{H} has a finite *probabilistic* bisimulation equivalence quotient. This result unifies and generalises previous results, and has the consequence that we can identify classes of probabilistic hybrid automata with a finite probabilistic bisimulation equivalence quotient on the basis of whether members of the corresponding class of hybrid automata have finite bisimulation quotients. This automatically extends the set of classes of probabilistic hybrid automata for which a finite bisimulation equivalence quotient exists (for example, given the existence of finite bisimulation equivalence quotients for STORMED hybrid automata [VPVD08], we can conclude that probabilistic hybrid automata to which the restrictions of STORMED hybrid automata apply to the non-probabilistic characteristics of the system have a finite number of probabilistic bisimulation equivalence classes). Any future results on the identification of classes of hybrid automata with finite bisimulation quotients will also imply that the corresponding class of probabilistic hybrid automata has finite probabilistic bisimulation quotients.

We also consider a particular example of the application of this result, namely probabilistic rectangular automata with a discrete-time semantics. Rectangular automata [HKPV98] are a subclass of hybrid automata with both interesting theoretical properties and practical applications. In a rectangular automaton, the continuous dynamics are governed by inclusions of the form $\dot{x} \in I$, where I is an interval. The motivation for such inclusions is that they can over-approximate complex continuous dynamics [HHWT98, DHR05]. However, even simple verification problems for rectangular automata, such as determining whether an error state is reachable, are undecidable [ACH⁺95, HKPV98]. In [HK99], a *discrete-time* assumption requires that jumps between nodes can only occur at evenly-spaced points in time. In this paper we consider the application of the discrete-time assumption to probabilistic rectangular automata: by our results, the existence of a computable finite bisimulation equivalence in the non-probabilistic setting implies the existence of a computable finite probabilistic bisimulation equivalence in the probabilistic setting, in turn showing that verification and control problems for probabilistic rectangular automata with a discrete-time semantics are decidable.

After introducing some preliminary concepts in Section 2 and probabilistic hybrid automata in Section 3 (some of which, such as considering ω -regular properties for PHA, are novel to this paper), we relate non-probabilistic and probabilistic bisimulation on non-probabilistic and probabilistic hybrid automata in Section 4. In Section 5, we apply the result to discrete-time probabilistic rectangular automata. We consider control and verification with respect to the class of ω -regular properties, modeled here as deterministic Rabin or Streett automata, which allow us to specify a wide variety of safety and liveness requirements.

Related work Previous work in the field of probabilistic rectangular automata has considered mainly dense-time verification problems for the subclass of probabilistic timed automata, in which continuous dynamics are of the form $\dot{x} = 1$ for all variables x in all locations, and with respect to properties expressed in the probabilistic temporal logic PTCTL [KNSS02]. The dense-time verification problem for probabilistic timed automata is EXPTIME-complete both for probabilistic temporal logic properties [KNSS02, LS07] and for ω -regular properties [Spr11]. The discrete-time verification problem for probabilistic timed automata is also EXPTIME-complete [KNPS06]: in the case of probabilistic timed automata, the discrete-time semantics corresponds directly to a finite-state system in which variables take natural-numbered values only, which is not the case for probabilistic rectangular automata due to the possibility of continuous nondeterministic choice in the continuous dynamics and in the resetting of variables. A dense-time controller synthesis problem concerning the computation of controllers of probabilistic timed automata that optimise the expected time to reach a state set has been considered in [FKNT16, JKNP17].

The dense-time verification problem for the class of *initialised* probabilistic rectangular automata, in which the condition on the continuous dynamics of a variable cannot be different before and after taking a probabilistic edge if the variable is not reset on the edge, with respect to reachability or safety objectives, has been considered in [Spr00, Spr01]. It is shown how a probabilistic rectangular automaton can be translated to a probabilistic timed automaton in which each variable is represented by two clocks, one clock representing an upper bound on the value of the variable, the other clock representing a

lower bound on the value of the variable, following the construction in the non-probabilistic setting of [OSY94, HKPV98]. While, in the non-probabilistic setting, the translation from an initialised rectangular automaton to a timed automaton preserves ω -regular properties, the translation from an initialised probabilistic rectangular automaton to a probabilistic timed automaton presented in [Spr00, Spr01] results in an abstract, approximate model. Similarly, [ZSR⁺12] presents an approach to the dense-time verification problem based on approximation, but for probabilistic hybrid automaton models with a form of non-rectangular continuous dynamics incomparable to that of rectangular automata. This work is extended in [HNP⁺11] to consider also control problems, and to consider an iterative refinement approach, in which the degree of approximation can be reduced by refining the finite-state abstraction of a probabilistic hybrid automaton. Other related work considering approximation includes the use of probabilistic hybrid automata as approximate models of stochastic hybrid automata, in which variables can be reset according to continuous probability distributions [FHH⁺11, Hah13].

A preliminary version of some results of this work can be found in [Spr11, Spr14]. In this paper we replace the *ad hoc* technical material developed for probabilistic rectangular automata in [Spr11] with the general framework of [Spr14] that shows that the existence of a finite non-probabilistic bisimulation relation can be used to show the existence of a probabilistic bisimulation relation. Furthermore, we show that the results of [Spr14] can be used not just for verification but also for control.

2 Preliminaries

We use \mathbb{R} to denote the set of real numbers, $\mathbb{R}_{\geq 0}$ to denote the set of non-negative real numbers, \mathbb{N} to denote the set of natural numbers, \mathbb{Z} to denote the set of integers, \mathbb{Q} to denote the set of rational numbers, and AP to denote a set of atomic propositions. Given a set Q and a function $\mu : Q \rightarrow \mathbb{R}_{\geq 0}$, we define $\text{support}(\mu) = \{q \in Q \mid \mu(q) > 0\}$. A (discrete) probability *distribution* over a countable set Q is a function $\mu : Q \rightarrow [0, 1]$ such that $\sum_{q \in Q} \mu(q) = 1$. Let $\text{Dist}(Q)$ be the set of distributions over Q . If Q is an uncountable set, we define $\text{Dist}(Q)$ to be the set of functions $\mu : Q \rightarrow [0, 1]$, such that $\text{support}(\mu)$ is a countable set and μ restricted to $\text{support}(\mu)$ is a (discrete) probability distribution. Given a set $Q' \subseteq Q$, we let $\mu[Q'] = \sum_{q \in Q'} \mu(q)$.

2.1 Probabilistic Games and Markov Decision Processes

A *probabilistic game* (or $2\frac{1}{2}$ -player game) $G = (S, \rightarrow, Lab)$ comprises the following components: a (possibly uncountable) set of *states* S ; a (possibly uncountable) *probabilistic transition relation* $\rightarrow \subseteq S \times 2^{\text{Dist}(S)} \setminus \emptyset$; and a *labeling function* $Lab : S \rightarrow 2^{AP}$. The transitions from state to state of a $2\frac{1}{2}$ -player game are performed in three steps: given that the current state is s , the first step concerns a nondeterministic selection by player 1 of $(s, \Lambda) \in \rightarrow$; the second step comprises a nondeterministic selection by player 2 of some $\mu \in \Lambda$; the third step comprises a probabilistic choice, made according to the distribution μ , as to which state to make the transition (that is, we then make a transition to a state $s' \in S$ with probability $\mu(s')$). Underlying this formulation of probabilistic games is the assumption that turns of the game are played in a cyclic manner, where each cycle consists first of the turn of player 1, then that of player 2, followed by that of the probabilistic player. This suffices for our purposes, but is in contrast to the usual presentation of $2\frac{1}{2}$ -player games (see, for example, [CH12]), in which the order of the turns of the game does not follow a fixed cycle. A $2\frac{1}{2}$ -player game is *total* if, for each state $s \in S$, there exists at least one transition $(s, \cdot) \in \rightarrow$. We generally consider total $2\frac{1}{2}$ -player games in this paper. Occasionally we omit the labeling function Lab for $2\frac{1}{2}$ -player games.

An *infinite path* of a $2\frac{1}{2}$ -player game G is an infinite sequence $r = s_0 \Lambda_0 \mu_0 s_1 \Lambda_1 \mu_1 \dots$ such that $(s_i, \Lambda_i) \in \rightarrow$, $\mu_i \in \Lambda_i$ and $\mu_i(s_{i+1}) > 0$ for each $i \in \mathbb{N}$. Similarly, a *finite path* of G is a finite sequence $r = s_0 \Lambda_0 \mu_0 s_1 \Lambda_1 \mu_1 \dots \Lambda_{n-1} \mu_{n-1} s_n$ such that $(s_i, \Lambda_i) \in \rightarrow$, $\mu_i \in \Lambda_i$ and $\mu_i(s_{i+1}) > 0$ for each $i < n$. If r is finite, the length of r , denoted by $|r|$, is equal to the number of transitions (subsequences of the form $s \Lambda \mu$) along r . If r is infinite, we let $|r| = \infty$. We use $IPath^G$ to denote the set of infinite paths of G , and $FPath^G$ to denote the set of finite paths of G . Let $IPath^G(s)$ and $FPath^G(s)$ refer to the sets of infinite and finite paths of G , respectively, commencing in state $s \in S$. When clear from the context we omit the superscript G . If r is a finite path, we denote by $last(r)$ the final state of r . For any path r and $i \leq |r|$, let $r(i) = s_i$ be the $(i+1)$ th state along r , let $dset(r, i) = \Lambda_i$ be the $(i+1)$ th distribution set featured along r and let $d(r, i) = \mu_i$ be the $(i+1)$ th distribution taken along r .

We now consider the notion of strategy which, for each player, specifies how the next transition should be chosen, given a finite execution history. Let $G = (S, \rightarrow, Lab)$ be a $2\frac{1}{2}$ -player game. A *player 1 strategy* of G is a function $\sigma : FPath \rightarrow \text{Dist}(\rightarrow)$ such that, for all finite paths $r \in FPath$, we have $\text{support}(\sigma(r)) \subseteq \{(s, \Lambda) \in \rightarrow \mid s = \text{last}(r)\}$ (i.e., after the finite history r , player 1 strategy σ assigns positive probability only to those transitions from the final state of r). Similarly, a *player 2 strategy* of G is a function $\pi : FPath \cdot 2^{\text{Dist}(S)} \rightarrow \text{Dist}(\text{Dist}(S))$ such that, for all finite paths $r \in FPath$ and set $\Lambda \in 2^{\text{Dist}(S)}$, we have $\text{support}(\pi(r\Lambda)) \subseteq \Lambda$ (i.e., after the finite history $r\Lambda$, player 2 strategy π assigns positive probability only to those distributions from Λ). In this paper (following the precedent of [Hah13] in the verification setting), we restrict our attention to strategies that choose according to distributions having finite support: that is, for $r \in FPath$ and $\Lambda \in 2^{\text{Dist}(S)}$, we have that $\text{support}(\sigma(r))$ and $\text{support}(\pi(r\Lambda))$ are finite. We write Σ_G and Π_G for the set of strategies of player 1 and player 2, respectively, on G . A pair $(\sigma, \pi) \in \Sigma_G \times \Pi_G$ is called a *strategy profile*. For any strategy profile (σ, π) , let $IPath^{\sigma, \pi}$ and $FPath^{\sigma, \pi}$ denote the sets of infinite and finite paths, respectively, resulting from the choices of (σ, π) : for example, $IPath^{\sigma, \pi} = \{r \in IPath \mid \forall i \in \mathbb{N}. dset(r, i) \in \text{support}(\sigma(r(i))) \text{ and } d(r, i) \in \text{support}(\pi(r(i)dset(r, i)))\}$. For a state $s \in S$, let $IPath^{\sigma, \pi}(s) = IPath^{\sigma, \pi} \cap IPath(s)$ and $FPath^{\sigma, \pi}(s) = FPath^{\sigma, \pi} \cap FPath(s)$. Given a strategy profile $(\sigma, \pi) \in \Sigma_G \times \Pi_G$ and a state $s \in S$, we define the probability measure $Prob_s^{\sigma, \pi}$ over $IPath^{\sigma, \pi}(s)$ in the standard way (see, for example, [CH12]).

Given an infinite path $r = s_0\Lambda_0\mu_0s_1\Lambda_1\mu_1 \dots$ of a $2\frac{1}{2}$ -player game $G = (S, \rightarrow, Lab)$, the *trace* of r , denoted by $\text{trace}(r)$, is defined to be the infinite sequence $Lab(s_0)Lab(s_1) \dots$. Let $\text{Trace}(G)$ be the set of all traces of G (i.e., $\text{Trace}(G) = \{\text{trace}(r) \in (2^{AP})^\omega \mid r \in IPath^G\}$). An *objective* φ for G is a set of traces of G (i.e., $\varphi \subseteq \text{Trace}(G)$). In this paper, we will consider the class of ω -regular objectives. Given the ω -regular objective φ , a state $s \in S$ and a strategy profile (σ, π) , the set $\{r \in IPath^{\sigma, \pi}(s) \mid \text{trace}(r) \in \varphi\}$ is measurable (this follows from the fact that there is a countably-infinite-state Markov chain associated with a strategy profile (σ, π) and from Remark 10.57 of [BK08]). For simplicity we write $Prob_s^{\sigma, \pi}(\varphi)$ instead of $Prob_s^{\sigma, \pi}(\{r \in IPath^{\sigma, \pi}(s) \mid \text{trace}(r) \in \varphi\})$. The *value function* (for the property φ) is defined as the function $Val^G(\varphi)$ such that, for each state $s \in S$:

$$Val^G(\varphi)(s) = \sup_{\sigma \in \Sigma} \inf_{\pi \in \Pi} Prob_s^{\sigma, \pi}(\varphi).$$

A *Markov decision process* (MDP) is a $2\frac{1}{2}$ -player game (S, \rightarrow, Lab) for which $|\Lambda| = 1$ for each $(s, \Lambda) \in \rightarrow$. Usually we write the transition relation \rightarrow of an MDP as $\rightarrow \subseteq S \times \text{Dist}(S)$. In contrast to $2\frac{1}{2}$ -player games, the transitions from state to state of an MDP are performed in two steps: given that the current state is s , the first step concerns a nondeterministic selection of $(s, \mu) \in \rightarrow$; the second step comprises a probabilistic choice made according to the distribution μ . A (finite or infinite) path of an MDP is defined as for a $2\frac{1}{2}$ -player game, with only minor notational differences: for example, an infinite path of an MDP is denoted by $s_0\mu_0s_1\mu_1 \dots$, where $(s_i, \mu_i) \in \rightarrow$ and $\mu_i(s_{i+1}) > 0$ for each $i \in \mathbb{N}$. In the case of MDPs, player 2 has a trivial choice over a single element, and hence has only one strategy (i.e., $|\Pi| = 1$): therefore we use the term strategy to refer both to player 1 strategies and strategy profiles. Similarly, we omit the notation referring to the player 2 strategy, and write, for example, $IPath^\sigma(s)$ and $Prob_s^\sigma$. The value function for the MDP M is defined as $Val^M(\varphi)(s) = \sup_{\sigma \in \Sigma} Prob_s^\sigma(\varphi)$ for each state $s \in S$.

Given a $2\frac{1}{2}$ -player game $G = (S, \rightarrow, Lab)$ and state $s \in S$, objective φ and $\lambda \in \mathbb{Q} \cap (0, 1]$, the *game-based threshold problem* for G, s, φ, λ consists of deciding whether $Val^G(\varphi)(s) \geq \lambda$. Analogously, given state s of MDP M , the *MDP threshold problem* for M, s, φ, λ consists of deciding whether $Val^M(\varphi)(s) \geq \lambda$. We note that, for finite $2\frac{1}{2}$ -player games and ω -regular properties described as deterministic Rabin (Streett, respectively) automata, the game-based threshold problem is in NP (coNP, respectively) [CdAH05]; instead, for finite MDPs and ω -regular properties described as deterministic Rabin or Streett automata, the MDP threshold problem is solvable in polynomial time [dA97, CdAH05].

2.2 Probabilistic bisimulation

Let $G = (S, \rightarrow, Lab)$ be a $2\frac{1}{2}$ -player game, and let $\equiv \subseteq S \times S$ be an equivalence relation on S . We lift \equiv to equivalence relations on $\text{Dist}(S)$ and $2^{\text{Dist}(S)}$ in the following way. For distributions $\mu, \nu \in \text{Dist}(S)$, we denote by $\mu \equiv \nu$ the condition that, for each equivalence class C of \equiv , the equality $\mu[C] = \nu[C]$ holds. Furthermore, for sets $\Lambda, \Xi \subseteq \text{Dist}(S)$ of distributions, we use $\Lambda \equiv \Xi$ to denote that (1) for each $\mu \in \Lambda$, there exists $\nu \in \Xi$ such that $\mu \equiv \nu$, and (2) for each $\nu \in \Xi$, there exists $\mu \in \Lambda$ such that $\mu \equiv \nu$. A

probabilistic bisimulation respecting \equiv on \mathbf{G} [LS91, SL95, ZP10] is an equivalence relation $\approx \subseteq S \times S$ such that $s \approx t$ implies that (1) $s \equiv t$, and (2) if $(s, \Lambda) \in \rightarrow$, then there exists $(t, \Xi) \in \rightarrow$ such that $\Lambda \approx \Xi$. Recalling that the transition relation for MDPs can be written as $\rightarrow \subseteq S \times \text{Dist}(S)$, we can slightly modify condition (2) to obtain probabilistic bisimulation for MDPs in the following way: (2) if $(s, \mu) \in \rightarrow$, then there exists $(t, \nu) \in \rightarrow$ such that $\mu \approx \nu$. Let $\mathbf{G} = (S, \rightarrow, \text{Lab})$ be a $2\frac{1}{2}$ -player game and let $\equiv_{\mathbf{G}}^{\text{lab}}$ be the smallest equivalence relation on S such that $s \equiv_{\mathbf{G}}^{\text{lab}} t$ if $\text{Lab}(s) = \text{Lab}(t)$, for each $s, t \in S$. In the sequel, we generally make reference to probabilistic bisimulations respecting $\equiv_{\mathbf{G}}^{\text{lab}}$.

Proposition 1. *Let $\mathbf{G} = (S, \rightarrow, \text{Lab})$ be a $2\frac{1}{2}$ -player game, φ be an ω -regular objective, \approx be a probabilistic bisimulation respecting $\equiv_{\mathbf{G}}^{\text{lab}}$, and $s, t \in S$ be such that $s \approx t$. Then $\text{Val}^{\mathbf{G}}(\varphi)(s) = \text{Val}^{\mathbf{G}}(\varphi)(t)$.*

Proof. (Sketch.) By definition of $\text{Val}^{\mathbf{G}}(\varphi)$, we need to show that:

$$\sup_{\sigma \in \Sigma} \inf_{\pi \in \Pi} \text{Prob}_s^{\sigma, \pi}(\varphi) = \sup_{\sigma \in \Sigma} \inf_{\pi \in \Pi} \text{Prob}_t^{\sigma, \pi}(\varphi).$$

Given that \approx is an equivalence, it suffices to show that $\sup_{\sigma \in \Sigma} \inf_{\pi \in \Pi} \text{Prob}_s^{\sigma, \pi}(\varphi) \leq \sup_{\sigma \in \Sigma} \inf_{\pi \in \Pi} \text{Prob}_t^{\sigma, \pi}(\varphi)$. To establish this, we show that, for every $\sigma \in \Sigma$, there exists $\sigma' \in \Sigma$ such that, for every $\pi' \in \Pi$, there exists $\pi \in \Pi$ such that $\text{Prob}_s^{\sigma, \pi}(\varphi) = \text{Prob}_t^{\sigma', \pi'}(\varphi)$. The proof of this fact can be derived as a special case of Lemma 8 of [ZP10], which corresponds to a game-based version of the “execution correspondence theorem” of [Seg95]. Intuitively, the proof proceeds by showing that, for every $\sigma \in \Sigma$ and $\pi' \in \Pi$, there exist $\sigma' \in \Sigma$ and $\pi \in \Pi$ such that the Markov chains obtained from strategy profile (σ, π) and from (σ', π') are probabilistically bisimilar from their initial states s and t , respectively. Note that φ does not distinguish between different paths which visit the same sequences of probabilistic bisimulation equivalence classes, thus implying that the ω -regular objectives that we consider are “bisimulation closed” in the terminology of [BK08]. Then, from this fact and Lemma 10.66 of [BK08], we have that $\text{Prob}_s^{\sigma, \pi}(\varphi) = \text{Prob}_t^{\sigma', \pi'}(\varphi)$. \square

Consider a probabilistic bisimulation \approx that respects $\equiv_{\mathbf{G}}^{\text{lab}}$. Let \mathfrak{C} be the set of equivalence classes of \approx . Given a set $\Lambda \subseteq \text{Dist}(S)$ of distributions over states of \mathbf{G} , then we let $\mathbf{qdist}(\Lambda) \subseteq \text{Dist}(\mathfrak{C})$ be the set of distributions over equivalence classes of \approx such that, for each distribution μ in Λ , there exists a distribution $\nu \in \mathbf{qdist}(\Lambda)$ such that the probability assigned to a particular equivalence class by ν is the sum of the the probabilities assigned to states of that class by μ . Formally, we let $\mathbf{qdist}(\Lambda) = \{\nu \in \text{Dist}(\mathfrak{C}) \mid \exists \mu \in \Lambda. \forall C \in \mathfrak{C}. \nu(C) = \sum_{s \in C} \mu(s)\}$. Then the *quotient* of \mathbf{G} and the probabilistic bisimulation \approx is the $2\frac{1}{2}$ -player game $\mathfrak{Q}[\mathbf{G}] = (\mathfrak{C}, \rightsquigarrow, \mathfrak{Lab})$ where the set of states is equal to the set \mathfrak{C} of equivalence classes of \approx , and where:

- \rightsquigarrow is the smallest set of transitions such that, for each $C \in \mathfrak{C}$, $s \in C$ and $(s, \Lambda) \in \rightarrow$, we have $(C, \mathbf{qdist}(\Lambda)) \in \rightsquigarrow$;
- \mathfrak{Lab} is defined by $\mathfrak{Lab}(C) = \text{Lab}(s)$, for each $C \in \mathfrak{C}$ and an arbitrary $s \in C$.

The labelling condition \mathfrak{Lab} is well-defined because \approx respects $\equiv_{\mathbf{G}}^{\text{lab}}$.

Given $2\frac{1}{2}$ -player games $\mathbf{G}_1 = (S_1, \rightarrow_1, \text{Lab}_1)$ and $\mathbf{G}_2 = (S_2, \rightarrow_2, \text{Lab}_2)$, we let the *union* $2\frac{1}{2}$ -player game be defined by $\mathbf{G}_1 \uplus \mathbf{G}_2 = (S_1 \uplus S_2, \rightarrow_1 \uplus \rightarrow_2, \text{Lab})$ where $\text{Lab}(s) = \text{Lab}_1(s)$ if $s \in S_1$ and $\text{Lab}(s) = \text{Lab}_2(s)$ if $s \in S_2$.

Lemma 1. *Let $\mathbf{G} = (S, \rightarrow, \text{Lab})$ be a $2\frac{1}{2}$ -player game, let $s \in S$ be a state of \mathbf{G} , let \approx be a probabilistic bisimulation respecting $\equiv_{\mathbf{G}}^{\text{lab}}$ on \mathbf{G} , and let C denote the equivalence class of \approx such that $s \in C$. Then $\text{Val}^{\mathbf{G}}(\varphi)(s) = \text{Val}^{\mathfrak{Q}[\mathbf{G}]}(\varphi)(C)$.*

Proof. (Sketch.) Let \approx' be an equivalence relation on $S \cup \mathfrak{C}$ defined as the smallest equivalence relation such that $s \approx' C$ if $s \in C$. Then, as a consequence of the definition of $\mathfrak{Q}[\mathbf{G}]$ and $\mathbf{G} \uplus \mathfrak{Q}[\mathbf{G}]$, we have that \approx' is a probabilistic bisimulation respecting $\equiv_{\mathbf{G} \uplus \mathfrak{Q}[\mathbf{G}]}^{\text{lab}}$ on $\mathbf{G} \uplus \mathfrak{Q}[\mathbf{G}]$. Then the result follows from Proposition 1. \square

Given that MDPs are a subclass of $2\frac{1}{2}$ -player games, the corresponding results for MDPs can be obtained in a similar manner. That is, for an MDP $\mathbf{M} = (S, \rightarrow, \text{Lab})$, for states $s, t \in S$ related by a probabilistic bisimulation \approx respecting $\equiv_{\mathbf{M}}^{\text{lab}}$, we have $\text{Val}^{\mathbf{M}}(\varphi)(s) = \text{Val}^{\mathbf{M}}(\varphi)(t)$. Furthermore, for $s \in S$ and equivalence class C of \approx such that $s \in C$, we have $\text{Val}^{\mathbf{M}}(\varphi)(s) = \text{Val}^{\mathfrak{Q}[\mathbf{M}]}(\varphi)(C)$. Note that the quotient of an MDP is itself an MDP.

2.3 Probabilistic labelled transition systems

We now introduce probabilistic labelled transition systems, which will be used in subsequent sections to define the semantics of probabilistic hybrid automata. A *probabilistic labeled transition system* (PLTS) $T = (S, Act, \Rightarrow, Lab)$ comprises the following components: a (possibly uncountable) set of states S ; a (possibly uncountable) set of actions Act ; a (possibly uncountable) transition relation $\Rightarrow \subseteq S \times Act \times \text{Dist}(S)$; and a labeling function $Lab : S \rightarrow 2^{AP}$. The notions of totality and paths of PLTSs are adapted in a straightforward way from $2\frac{1}{2}$ -player games: for example, an infinite path of a PLTS is denoted by $r = s_0 a_0 \mu_0 s_1 a_1 \mu_1 \dots$ where $a_i \in Act$ and $s_{i+1} \in \text{support}(\mu_i)$ for each $i \in \mathbb{N}$.

We can interpret the underlying model of a PLTS in one of two ways: as a $2\frac{1}{2}$ -player game, for the control setting, or as an MDP, for the verification setting.

Control: Player 1 (the controller) chooses which action should be taken and player 2 (the environment) chooses the exact transition that is then executed, provided that it corresponds to the action chosen by player 1. Formally, the $2\frac{1}{2}$ -player game interpretation of T is a $2\frac{1}{2}$ -player game $G(T) = (S, \rightarrow, Lab)$ where \rightarrow is the smallest set such that, for each $s \in S$ and $a \in Act$ such that there exists $(s, a, \mu) \in \Rightarrow$, we have $(s, \{\mu \mid (s, a, \mu) \in \Rightarrow\}) \in \rightarrow$.

Verification: All of the nondeterministic choices of which transitions to take are under the control of a single player. The MDP interpretation of T is an MDP $M(T) = (S, \rightarrow, Lab)$ where \rightarrow is the smallest set such that $(s, a, \mu) \in \Rightarrow$ implies $(s, \mu) \in \rightarrow$.

Let $T = (S, Act, \Rightarrow, Lab)$ be a PLTS, and let $\equiv \subseteq S \times S$ be an equivalence relation on S . A *probabilistic bisimulation respecting \equiv* on T [LS91, SL95] is an equivalence relation $\simeq \subseteq S \times S$ such that $s \simeq t$ implies that (1) $s \equiv t$, and (2) if $(s, a, \mu) \in \Rightarrow$, then there exists $(t, a, \nu) \in \Rightarrow$ such that $\mu \simeq \nu$. States s and t are called probabilistically bisimilar with respect to \equiv in T if there exists a probabilistic bisimulation \simeq respecting \equiv such that $s \simeq t$.

Lemma 2. Let $T = (S, Act, \Rightarrow, Lab)$ be a PLTS, let $\equiv \subseteq S \times S$ be an equivalence relation on S , and let \simeq be a probabilistic bisimulation respecting \equiv on T . Then \simeq is a probabilistic bisimulation respecting \equiv on $G(T)$ and $M(T)$.

Proof. Given that condition (1) in the definition of probabilistic bisimulation respecting \equiv is the same for T , $G(T)$ and $M(T)$, it remains to consider condition (2). We consider the case of $G(T)$ (the case of $M(T)$ is similar). Let $s, t \in S$ such that $s \simeq t$. By the definition of $G(T)$, if $(s, a, \mu) \in \Rightarrow$, then $(s, \Lambda) \in \rightarrow$ for $\Lambda = \{\mu \mid (s, a, \mu) \in \Rightarrow\}$. Given that $s \simeq t$, there exists $(t, a, \nu) \in \Rightarrow$ and, by the definition of $G(T)$, we have $(t, \Xi) \in \rightarrow$ for $\Xi = \{\nu \mid (t, a, \nu) \in \Rightarrow\}$. By the definition of probabilistic bisimulation respecting \equiv on T , we have that, for each $(s, a, \mu') \in \Rightarrow$, there exists $(t, a, \nu') \in \Rightarrow$ such that $\mu' \simeq \nu'$; this means that, for each $\mu' \in \Lambda$, there exists $\nu' \in \Xi$ such that $\mu' \simeq \nu'$. Given that \simeq is an equivalence, we can also conclude that, for each $\nu' \in \Xi$, there exists $\mu' \in \Lambda$ such that $\mu' \simeq \nu'$. Hence we have that $\Lambda \simeq \Xi$. This means that we have shown that condition (2) of the definition of probabilistic bisimulation on T implies condition (2) of the definition of probabilistic bisimulation on G . \square

Let \equiv_T^{lab} be the smallest equivalence relation on S such that $s \equiv_T^{lab} t$ if $Lab(s) = Lab(t)$, for each $s, t \in S$. The following corollary is a consequence of Proposition 1 and Lemma 2.

Corollary 1. Let $T = (S, Act, \Rightarrow, Lab)$ be a PLTS and let \simeq be a probabilistic bisimulation respecting \equiv_T^{lab} on T . Then for states $s, t \in S$ such that $s \simeq t$, we have $Val^{G(T)}(\varphi)(s) = Val^{G(T)}(\varphi)(t)$ and $Val^{M(T)}(\varphi)(s) = Val^{M(T)}(\varphi)(t)$.

A PLTS $T = (S, Act, \Rightarrow, Lab)$ for which all transitions $(s, a, \mu) \in \Rightarrow$ are such that μ is of the form $\{s' \mapsto 1\}$ for some $s' \in S$ is called a *nondeterministic labelled transition system* (NLTS). In the case of NLTSs, we often write (s, a, s') to denote the transition $(s, a, \{s' \mapsto 1\})$. If T is a NLTS, then we can simplify the definition of probabilistic bisimulation which, in this context, is called simply *bisimulation*, in the following way: a bisimulation respecting equivalence \equiv on a NLTS $(S, Act, \Rightarrow, Lab)$ is an equivalence relation $\approx \subseteq S \times S$ such that $s \approx t$ implies that (1') $s \equiv t$, and (2') if $(s, a, s') \in \Rightarrow$, then there exists $(t, a, t') \in \Rightarrow$ such that $s' \approx t'$. States s and t are called bisimilar with respect to \equiv in the NLTS if there exists a bisimulation \approx respecting \equiv such that $s \approx t$.

3 Probabilistic Hybrid Automata

In this section, we introduce probabilistic hybrid automata as a formal model for hybrid systems with discrete probabilistic choices. Let \mathcal{X} be a finite set of real-valued variables. A *valuation* $v : \mathcal{X} \rightarrow \mathbb{R}$ for \mathcal{X} is a function that assigns a value to each variable of \mathcal{X} . Let $\mathcal{V}(\mathcal{X})$ be the set of valuations for \mathcal{X} . When the set \mathcal{X} is clear from the context, we generally write \mathcal{V} .

A *probabilistic hybrid automaton* (PHA) $\mathcal{H} = (L, \mathcal{X}, \text{Events}, \text{post}, \text{prob}, \mathcal{L})$ consists of the following components:

- a finite set L of *locations*;
- a finite set \mathcal{X} of variables;
- a finite set *Events* of *events*;
- a *post operator* $\text{post} : L \times \mathcal{V} \times \mathbb{R}_{\geq 0} \rightarrow 2^{\mathcal{V}}$;
- a finite set $\text{prob} \subseteq L \times 2^{\mathcal{V}} \times \text{Events} \times \text{Dist}(\text{Upd}(\mathcal{X}) \times L)$ of *probabilistic edges*, where $\text{Upd}(\mathcal{X})$ is the set of *update functions* $\mathbf{u} : \mathcal{V} \rightarrow 2^{\mathcal{V}}$;
- a labelling function $\mathcal{L} : L \rightarrow 2^{AP}$.

A probabilistic edge $(l, g, e, p) \in \text{prob}$ comprises (1) a source location l , (2) a set g of valuations, called a *guard*, (3) an event e , and (4) a probability distribution p that assigns probability to pairs of the form (\mathbf{u}, l') , where $\mathbf{u} \in \text{Upd}(\mathcal{X})$ is a function describing the manner in which variables are updated and $l' \in L$ is a target location.

The behaviour of a PHA takes a similar form to that of a classical, non-probabilistic hybrid automaton [ACH⁺95]. If the PHA is currently in location l , as time passes, the values of the variables in \mathcal{X} change according to the post operator post : more precisely, if the current valuation is v and $\delta \in \mathbb{R}_{\geq 0}$ time units elapse, the valuation obtained after the elapse of time belongs to the set $\text{post}(l, v, \delta)$. If the current valuation of the variables belongs to the guard g of a probabilistic edge (l, g, e, p) , then that probabilistic edge can be taken. Taking a probabilistic edge (l, g, e, p) involves a probabilistic choice according to the distribution p : if this probabilistic choice selects the pair (\mathbf{u}, l') , then the PHA goes to location l' , updating the variables according to the function \mathbf{u} . More precisely, if the current valuation of the variables is v and the pair (\mathbf{u}, l') is chosen, then the state after taking the probabilistic edge will be (l', v') for some v' that is chosen nondeterministically from the set $\mathbf{u}(v)$. To summarise, the following choices made by the PHA are nondeterministic: the amount of time to let advance in the current location l ; the valuation used to describe the values of the variables after time has elapsed, according to post ; the probabilistic edge taken (provided that the guard of the probabilistic edge is satisfied by the current variable valuation); and, finally, the values to which the variables are updated when a probabilistic edge is taken. Instead, the only probabilistic choice featured in the model concerns the choice of pair (\mathbf{u}, l') once a probabilistic edge has been chosen, which is made according to a *discrete* probability distribution.

We make a number of standard assumptions on the components of a PHA [Spr01, Hah13].

- (*Assumptions on post.*) For all locations $l \in L$ and valuations $v \in \mathcal{V}$, we require the following: (1) $\text{post}(l, v, 0) = \{v\}$; (2) for all $\delta, \delta' \in \mathbb{R}_{\geq 0}$ such that $\delta \geq \delta'$, we have $\text{post}(l, v, \delta) = \bigcup_{v' \in \text{post}(l, v, \delta')} \text{post}(l, v', \delta - \delta')$.
- (*Probabilistic edges can be taken when no more time can elapse.*) If $l \in L$ and $v \in \mathcal{V}$ are such that $\text{post}(l, v, \delta) = \emptyset$ for all $\delta \in \mathbb{R}_{\geq 0}$ such that $\delta > 0$, then there must exist some probabilistic edge $(l, g, e, p) \in \text{prob}$ such that $v \in g$.
- (*Non-empty updates.*) For all probabilistic edges $(l, g, e, p) \in \text{prob}$ and each $(\mathbf{u}, l') \in \text{support}(p)$, we have $\mathbf{u}(v) \neq \emptyset$ for all $v \in g$.
- (*Finite probabilistic branching.*) For all probabilistic edges $(l, g, e, p) \in \text{prob}$, the set $\text{support}(p)$ is finite.

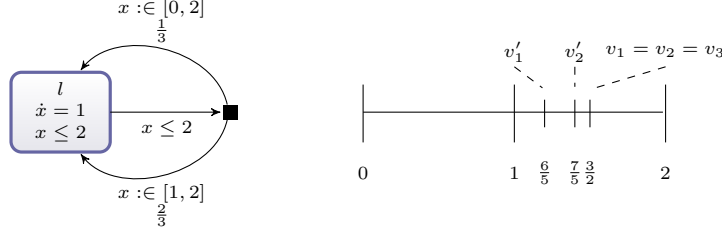


Figure 2: A PHA to illustrate the use of summation in the definition of PHA semantics

We now introduce formally the semantics of PHA in terms of PLTSs. The *(dense-time) semantics of the PHA* $\mathcal{H} = (L, \mathcal{X}, Events, post, prob, \mathcal{L})$ is the PLTS $\llbracket \mathcal{H} \rrbracket^{\text{dense}} = (S, Act, \Rightarrow, Lab)$ defined in the following way. The set of states of $\llbracket \mathcal{H} \rrbracket^{\text{dense}}$ is defined as $S = L \times \mathcal{V}$. The set of actions of $\llbracket \mathcal{H} \rrbracket^{\text{dense}}$ is $Act = \mathbb{R}_{\geq 0} \cup Events$. To define the transition relation \Rightarrow of $\llbracket \mathcal{H} \rrbracket^{\text{dense}}$, we first define sets of transitions corresponding to each time duration and event.

- *(Time elapse.)* Let $\delta \in \mathbb{R}_{\geq 0}$. Then $\xRightarrow{\delta} \subseteq S \times \mathbb{R}_{\geq 0} \times \text{Dist}(S)$ is the smallest set such that $((l, v), \delta, \{(l', v') \mapsto 1\}) \in \xRightarrow{\delta}$ if (1) $l = l'$ and (2) $v' \in post(l, v, \delta)$.
- *(Jumps.)* Let $e \in Events$. Consider a distribution $p \in \text{Dist}(Upd(\mathcal{X}) \times L)$, where $\text{support}(p) = \{(u_1, l_1), \dots, (u_n, l_n)\}$. Then, for valuation v , we write $\text{Bundle}(v, p) \subseteq \mathcal{V}^n$ to denote the smallest set of n -tuples of valuations such that $(v_1, \dots, v_n) \in \text{Bundle}(v, p)$ if $v_i \in u_i(v)$ for each $1 \leq i \leq n$. Then $\xRightarrow{e} \subseteq S \times \text{Dist}(S)$ is the smallest set of transitions such that $((l, v), e, \mu) \in \xRightarrow{e}$ if there exists a probabilistic edge $(l, g, e, p) \in prob$ such that (a) $v \in g$ and (b) there exists $(v_1, \dots, v_n) \in \text{Bundle}(v, p)$ such that, for each $(l', v') \in S$:

$$\mu(l', v') = \sum_{1 \leq i \leq n \text{ s.t. } v' = v_i} p(u_i, l').$$

The transition relation \Rightarrow of $\llbracket \mathcal{H} \rrbracket^{\text{dense}}$ is defined as the union of the transition sets defined above: formally, $\Rightarrow = (\bigcup_{\delta \in \mathbb{R}_{\geq 0}} \xRightarrow{\delta}) \cup (\bigcup_{e \in Events} \xRightarrow{e})$. Finally, we define the labelling function in the following way: for each $(l, v) \in S$, let $Lab(l, v) = \mathcal{L}(l)$.

Example 2. We note that the summation in the definition of jump transitions (i.e., in the definition of \xRightarrow{e} for $e \in Events$) is necessary for the case in which the same state can be obtained by more than one element (u, l) in the support set of the distribution of a probabilistic edge. Consider the PHA of Figure 2 (left). The PHA has a single location l , a single variable x , and a single probabilistic edge $(l, x \leq 2, e, p)$ with $p(u_{[0,2]}, l) = \frac{1}{3}$ and $p(u_{[1,2]}, l) = \frac{2}{3}$, where $u_{[0,2]}$ and $u_{[1,2]}$ are the update functions that assigns valuations in the sets $\{v \mid v(x) \in [0, 2]\}$ and $\{v \mid v(x) \in [1, 2]\}$, respectively, to any valuation. Consider valuation pair (v_1, v_2) and valuation v_3 that agree on the value assigned to x (for example, $v_1(x) = v_2(x) = v_3(x) = \frac{3}{2}$; this, and other valuations used later in this example are shown on Figure 2 (right)). Note that $(v_1, v_2) \in \text{Bundle}(v, p)$, where v denotes the valuation directly before taking the jump transition. Then the distribution μ associated with the valuation pair (v_1, v_2) according to the definition of the semantics of PHA is such that $\mu(l, v_3) = \frac{1}{3} + \frac{2}{3} = 1$. That is, the probability assigned to state (l, v_3) is obtained by summing the probabilities assigned to $(u_{[0,2]}, l)$ and $(u_{[1,2]}, l)$ by p . Now consider valuation pair $(v'_1, v'_2) \in \text{Bundle}(v, p)$ such that v'_1 and v'_2 disagree (for example $v'_1(x) = \frac{6}{5}$ and $v'_2(x) = \frac{7}{5}$). Then the distribution μ' associated with valuation pair (v'_1, v'_2) is such that $\mu(l, v'_1) = \frac{1}{3}$ and $\mu(l, v'_2) = \frac{2}{3}$: hence, in the case of (v'_1, v'_2) such that $v'_1 \neq v'_2$, we have that no summation is required (the probability of state (l, v'_1) is equal to the probability of $(u_{[0,2]}, l)$, and the probability of state (l, v'_2) is equal to the probability of $(u_{[1,2]}, l)$).

We consider two variants of the semantics of PHA, namely the time-abstract semantics, in which actions corresponding to durations of time-elapse transitions are replaced by a single action τ (where $\tau \notin Events$), and the discrete-time semantics, in which only time elapse transitions of duration 1 are represented. Formally, the *time-abstract semantics* of \mathcal{H} is the PLTS $\llbracket \mathcal{H} \rrbracket^{\text{ta}} = (S, Act, \Rightarrow, Lab)$, where the

set S of states and the labelling function Lab are the same as for the dense-time semantics of \mathcal{H} , the set of actions is defined as $Act = \{\tau\} \cup Events$, and the transition relation \Rightarrow is defined as $\xRightarrow{\tau} \cup (\bigcup_{e \in Events} \xRightarrow{e})$, where $\xRightarrow{\tau}$ is defined as $\{(s, \tau, \mu) \mid \exists \delta \in \mathbb{R}_{\geq 0}. (s, \delta, \mu) \in \xRightarrow{\delta}\}$. The *discrete-time semantics* of \mathcal{H} is the PLTS $\llbracket \mathcal{H} \rrbracket^{disc} = (S, Act, \Rightarrow, Lab)$ where the set S of states and the labelling function Lab are the same as for the dense-time semantics of \mathcal{H} , the set of actions is defined as $Act = \{1\} \cup Events$, and the transition relation \Rightarrow is defined as $\xRightarrow{1} \cup (\bigcup_{e \in Events} \xRightarrow{e})$. In the following, we let $\Delta^{dense} = \mathbb{R}_{\geq 0}$, $\Delta^{ta} = \{\tau\}$ and $\Delta^{disc} = \{1\}$. We consider the intuition underlying these definitions for control problems (with verification problems being more straightforward); given that the different semantics differ in terms of time-elapse transitions, we concentrate on these transitions. Consider the standard dense-time semantics: for a time-elapse transition, player 1 selects the time duration, and player 2 selects the exact transition, which encodes the exact element of *post* selected. Now consider the time-abstract semantics: player 1 selects the time elapse action τ ; then player 2 selects the exact transition, which encodes information on the actual duration elapsed (whereas the duration was chosen by player 1 in the dense-time semantics) and the exact element of *post* selected. Instead in the discrete-time semantics, player 1 selects the time elapse action 1 (because 1 is the only possible time duration), then player 2 selects the exact transition used on the basis of the selected element of *post*.

We now define the verification and control problems for PHA. Let φ be an objective, $\star \in \{\text{dense}, \text{ta}, \text{disc}\}$, let $\mathcal{H} = (L, \mathcal{X}, Events, post, prob, \mathcal{L})$ be a PHA with semantics $\llbracket \mathcal{H} \rrbracket^\star = (S, Act, \Rightarrow, Lab)$, let $s \in S$, and let $\lambda \in \mathbb{Q} \cap (0, 1]$. Then the *control problem* for $\mathcal{H}, s, \star, \varphi, \lambda$ returns YES if and only if the game-based threshold problem for $G(\llbracket \mathcal{H} \rrbracket^\star), s, \varphi, \lambda$ returns YES; similarly, the associated verification problem for $\mathcal{H}, s, \star, \varphi, \lambda$ returns YES if and only if the MDP threshold problem for $M(\llbracket \mathcal{H} \rrbracket^\star), s, \varphi, \lambda$ returns YES.

Specification of objectives of PHA using deterministic Rabin and Streett automata We next recall basic concepts concerning Rabin and Streett automata, which we use for the specification of ω -regular properties. Our notation is adapted from [BK08, BGC09].

A *deterministic ω -automaton* $\mathcal{A} = (Q, \text{Alph}, \delta, q_{\text{init}}, \text{Acc})$ consists of a finite set Q of automaton states, a finite alphabet Alph , a transition function $\delta : Q \times \text{Alph} \rightarrow Q$, an initial state $q_{\text{init}} \in Q$ and an acceptance condition $\text{Acc} \subseteq 2^Q \times 2^Q$. Let $\text{Acc} = \{(H_1, K_1), \dots, (H_n, K_n)\}$. A set $S' \subseteq Q$ is called *Rabin accepting* if there exists $1 \leq i \leq n$ such that $S' \cap H_i = \emptyset$ and $S' \cap K_i \neq \emptyset$. The set S' is called *Streett accepting* if for each $1 \leq i \leq n$ we have $S' \cap H_i \neq \emptyset$ or $S' \cap K_i = \emptyset$.

Let $\varsigma = v_1 v_2 v_3 \dots$ be an infinite word over Alph . The *run* for ς is the unique infinite sequence $\rho_\varsigma = q_0 q_1 q_2 \dots$ such that $q_0 = q_{\text{init}}$ and $q_i = \delta(q_{i-1}, v_i)$ for each $i \geq 1$. Let $\text{inf}(\rho_\varsigma)$ be the set of states that occur infinitely often along ρ_ς . Then the *Rabin-accepted language* of \mathcal{A} is $\text{Lang}_{\text{Rabin}}(\mathcal{A}) = \{\varsigma \in \text{Alph}^\omega \mid \text{inf}(\rho_\varsigma) \text{ is Rabin accepting}\}$. Similarly, the *Streett-accepted language* of \mathcal{A} is defined by $\text{Lang}_{\text{Streett}}(\mathcal{A}) = \{\varsigma \in \text{Alph}^\omega \mid \text{inf}(\rho_\varsigma) \text{ is Streett accepting}\}$. A *deterministic Rabin automaton* is a deterministic ω -automaton for which Rabin acceptance is used to define its language. Similarly, a *deterministic Streett automaton* is a deterministic ω -automaton for which Streett acceptance is used to define its language. In the following we use the alphabet $\text{Alph} = 2^{AP}$.

Let $\mathcal{H} = (L, \mathcal{X}, Events, post, prob, \mathcal{L})$ be a PHA and $\mathcal{A} = (Q, \text{Alph}, \delta, q_{\text{init}}, \text{Acc})$ be a deterministic ω -automaton. We define the *product PHA* $\mathcal{H} \otimes \mathcal{A} = (L \times Q, \mathcal{X}, Events, \widehat{post}, \widehat{prob}, \widehat{\mathcal{L}})$ as the PHA defined in the following way:

- $\widehat{post}((l, q), v, \delta) = post(l, v, \delta)$ for each $(l, q) \in L \times Q$, $v \in \mathcal{V}$ and $\delta \in \mathbb{R}_{\geq 0}$.
- \widehat{prob} is the smallest set of probabilistic edges such that $((l, q), g, e, \widehat{p}) \in \widehat{prob}$ if there exists $(l, g, e, p) \in prob$ such that:

$$\widehat{p}(u, (l', q')) = \begin{cases} p(u, l') & \text{if } q' = \delta(q, \mathcal{L}(l')) \\ 0 & \text{otherwise.} \end{cases}$$

- $\widehat{\mathcal{L}}(l, q) = \{q\}$ for each $(l, q) \in L \times Q$.

In the following, we restrict our attention to the case of the $2\frac{1}{2}$ -player game interpretation (the case of the MDP interpretation is similar). Hence, for $\star \in \{\text{dense}, \text{ta}, \text{disc}\}$, we consider $G(\llbracket \mathcal{H} \otimes \mathcal{A} \rrbracket^\star)$. For

$\circ \in \{\text{Rabin}, \text{Streett}\}$, we let accept_\circ be the set of traces of $G(\llbracket \mathcal{H} \otimes \mathcal{A} \rrbracket^\star)$ defined by $\text{accept}_\circ = \{\rho \in Q^\omega \mid \inf(\rho) \text{ is } \circ\text{-accepting}\}$.

Let (σ, π) be a strategy profile of $G(\llbracket \mathcal{H} \rrbracket^\star)$. Then we define the strategy profile (σ^+, π^+) of $G(\llbracket \mathcal{H} \otimes \mathcal{A} \rrbracket^\star)$ in the following way. First we note that, for any finite path $r = (l_0, v_0) \Lambda_0 \mu_0 (l_1, v_1) \Lambda_1 \mu_1 \cdots (l_{n-1}, v_{n-1}) \Lambda_{n-1} \mu_{n-1} (l_n, v_n)$ of $G(\llbracket \mathcal{H} \rrbracket^\star)$, there exists a unique path $r^+ = ((l_0, q_1), v_0) \Lambda'_0 \nu_0 \cdots ((l_{n-1}, q_n), v_{n-1}) \Lambda'_{n-1} \nu_{n-1} ((l_n, q_{n+1}), v_n)$ of $G(\llbracket \mathcal{H} \otimes \mathcal{A} \rrbracket^\star)$ where, for each $0 \leq i < n$ we have the following properties: (a) Λ_i and Λ' both correspond to either the time transition rule or the event transition rule for the same event, and (b) $\nu_i((l', \delta(q_{i-1}, \mathcal{L}(l'))), v') = \mu_i(l', v')$ for each $(l', v') \in S$ (it can be verified that such a distribution exists, if property (a) holds, by definition of $\mathcal{H} \otimes \mathcal{A}$). Vice versa, for any finite path r^+ of $G(\llbracket \mathcal{H} \otimes \mathcal{A} \rrbracket^\star)$, there exists a unique finite path r of $G(\llbracket \mathcal{H} \rrbracket^\star)$ satisfying the properties (a) and (b) above. Then the strategy σ^+ after path r^+ mimics the choice of σ after the path r : more precisely, if $\sigma(r) = a$, then $\sigma^+(r^+) = a$, for $a \in \Delta^* \cup \text{Events}$. Similarly, the strategy π^+ after path $r^+ \cdot \Lambda'$ mimics the choice of π after the path $r \cdot \Lambda$: that is, the strategy π^+ chooses from Λ' a distribution ν mimicking the choice of π of μ such that μ and ν satisfy the condition (b) above. It is also straightforward to see that we can obtain a strategy profile (σ, π) of $G(\llbracket \mathcal{H} \rrbracket^\star)$ from a strategy profile (σ^+, π^+) of $G(\llbracket \mathcal{H} \otimes \mathcal{A} \rrbracket^\star)$, by using similar reasoning.

The following result states the equality of the probability of a strategy profile (σ, π) exhibiting traces of $G(\llbracket \mathcal{H} \rrbracket^\star)$ accepted by \mathcal{A} with acceptance condition $\circ \in \{\text{Rabin}, \text{Streett}\}$ and the probability of the strategy profile (σ^+, π^+) exhibiting traces of $G(\llbracket \mathcal{H} \otimes \mathcal{A} \rrbracket^\star)$ that are \circ -accepting. Given that there is a one-to-one correspondence between strategy profiles (σ, π) of $G(\llbracket \mathcal{H} \rrbracket^\star)$ and (σ^+, π^+) of $G(\llbracket \mathcal{H} \otimes \mathcal{A} \rrbracket^\star)$, we omit the proof.

Proposition 2. *Let \mathcal{H} be a PHA, let \mathcal{A} be a deterministic ω -automaton with $\circ \in \{\text{Rabin}, \text{Streett}\}$ acceptance, let $(l, v) \in S$ be a state of $G(\llbracket \mathcal{H} \rrbracket^\star)$, and let (σ, π) be a strategy profile of $G(\llbracket \mathcal{H} \rrbracket^\star)$. Then:*

$$\text{Prob}_{(l,v)}^{\sigma,\pi}(\text{Lang}_\dagger(\mathcal{A})) = \text{Prob}_{((l,\delta(q_{\text{init}}, \text{Lab}(l))),v)}^{\sigma^+,\pi^+}(\text{accept}_\circ).$$

The proposition then implies that the problem of computing $\text{Val}^{G(\llbracket \mathcal{H} \rrbracket^\star)}(\text{Lang}_\dagger(\mathcal{A}))(s)$, for any state $s \in S$, can be reduced to that of computing $\text{Val}^{G(\llbracket \mathcal{H} \otimes \mathcal{A} \rrbracket^\star)}(\text{accept}_\circ)(s)$. By Lemma 1, we have $\text{Val}^{G(\llbracket \mathcal{H} \otimes \mathcal{A} \rrbracket^\star)}(\text{accept}_\circ)(s) = \text{Val}^{\Omega[G(\llbracket \mathcal{H} \otimes \mathcal{A} \rrbracket^\star)]}(\text{accept}_\circ)(C)$ for the unique \approx -equivalence class C for which $s \in C$. In the case in which (1) $\mathcal{H} \otimes \mathcal{A}$ has a finite number of probabilistic bisimulation equivalence classes that can be effectively computed, and (2) the probabilities of all distributions of $\Omega[G(\llbracket \mathcal{H} \otimes \mathcal{A} \rrbracket^\star)]$ are rational, the value function $\text{Val}^{\Omega[G(\llbracket \mathcal{H} \otimes \mathcal{A} \rrbracket^\star)]}(\text{accept}_\circ)$ can be computed using standard methods for computing value functions for Rabin and Streett acceptance conditions on finite-state $2\frac{1}{2}$ -player games [CdAH05]. In the MDP case, we can compute $\text{Val}^{\Omega[M(\llbracket \mathcal{H} \otimes \mathcal{A} \rrbracket^\star)]}(\text{accept}_\circ)$ using methods for computing value functions for Rabin and Streett acceptance on finite-state MDPs [dA97, CdAH05].

4 Bisimulations of Probabilistic Hybrid Automata

In this section we will consider the problem of reasoning about probabilistic bisimulation relations of probabilistic hybrid automata. In particular, we are interested in showing that previous results showing the existence of *non-probabilistic* bisimulation relations for *non-probabilistic* hybrid automata can be lifted to the probabilistic case.

A (*non-probabilistic*) *hybrid automaton* (HA) is a PHA $(L, \mathcal{X}, \text{Events}, \text{post}, \text{prob}, \mathcal{L})$ for which probabilistic edges $(l, g, e, p) \in \text{prob}$ correspond to a trivial probabilistic choice over a single element; more precisely, each $(l, g, e, p) \in \text{prob}$ is such that p is of the form $\{(u, l') \mapsto 1\}$ for some $u \in \text{Upd}(\mathcal{X})$ and $l' \in L$. We refer to probabilistic edges of the above form as *edges*. It can be observed that the semantics of an HA \mathcal{H} , namely $\llbracket \mathcal{H} \rrbracket^\star$ for any $\star \in \{\text{dense}, \text{ta}, \text{disc}\}$, is an NLTS.

Consider an arbitrary PHA $\mathcal{H} = (L, \mathcal{X}, \text{Events}, \text{post}, \text{prob}, \mathcal{L})$. In the following, we use $\text{ind}(\text{prob})$ to denote the edges induced by the probabilistic edges in prob : formally, let $\text{ind}(\text{prob})$ be the smallest set of edges such that, if $(l, g, e, p) \in \text{prob}$ then, for each $(u, l') \in \text{support}(p)$, there exists the edge $(l, g, (e, p, u), \{(u, l') \mapsto 1\}) \in \text{ind}(\text{prob})$. Furthermore, let $\text{ind}(\text{Events})$ be the set of events (triples of events, probabilistic edges and update functions) corresponding to edges in $\text{ind}(\text{prob})$: formally, let $\text{ind}(\text{Events}) = \{(e, p, u) \in \text{Events} \times \text{Dist}(\text{Upd}(\mathcal{X}) \times L) \times \text{Upd}(\mathcal{X}) \mid (l, g, (e, p, u), \{(u, l') \mapsto 1\}) \in \text{ind}(\text{prob})\}$. Now let $\text{ind}(\mathcal{H}) = (L, \mathcal{X}, \text{ind}(\text{Events}), \text{post}, \text{ind}(\text{prob}), \mathcal{L})$ be the HA *induced* by the PHA \mathcal{H} .

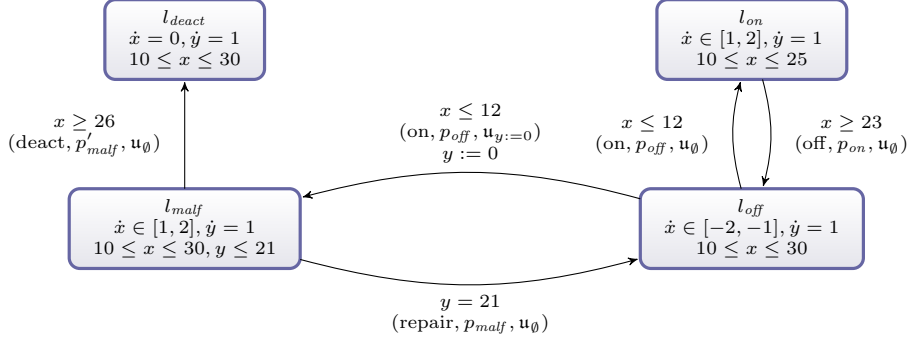


Figure 3: The induced HA of the PHA of Figure 1

Note that the location, variable and post sets, and the labelling function, are the same in \mathcal{H} and $\text{ind}(\mathcal{H})$. The set of edges of $\text{ind}(\mathcal{H})$ is $\text{ind}(\text{prob})$, i.e., a set of edges in which the events encode information derived from probabilistic edges in prob .

Example 3. The induced HA of the PHA of Figure 1 is shown in Figure 3. We use p_{on} and p_{off} to denote the distributions corresponding to probabilistic edges available from l_{on} and l_{off} , respectively. Furthermore, we use p_{malf} and p'_{malf} to denote the two distributions corresponding to probabilistic edges available from l_{on} and leading to l_{off} and l_{deact} , respectively. The update function u_0 denotes the identity function (i.e., the update does not change the value of either variable), whereas $u_{y:=0}$ denotes the function that resets the value of y to 0 and leaves the value of x unchanged. Each edge of the induced HA mimics one particular outcome of a probabilistic edge of the PHA. For example, the edge from l_{off} to l_{malf} shares the same source location and guard as the unique probabilistic edge $(l_{\text{off}}, x \leq 12, \text{on}, p_{\text{off}})$ from l_{off} , and has the same update function and target location as one of the probabilistic edge's outcomes, namely the pair $(u_{y:=0}, l_{\text{malf}})$. Furthermore, the event labelling the edge is a triple encoding not only the PHA event of the probabilistic edge, but also the distribution of the probabilistic edge and the update function of the outcome: this encoding allows us to recover information about the probabilistic branching structure of the PHA from the induced HA.

We now show that bisimilar states of the semantics of $\text{ind}(\mathcal{H})$ are probabilistically bisimilar in the semantics of \mathcal{H} . The equivalence relation $\equiv_{\text{loc}} \subseteq S \times S$ is defined as the smallest set such that $(l, v) \equiv_{\text{loc}} (m, w)$ if $l = m$, for all states $(l, v), (m, w) \in S$.

Proposition 3. Let $\star \in \{\text{dense}, \text{ta}, \text{disc}\}$ and let \simeq be a bisimulation with respect to \equiv_{loc} on $\llbracket \text{ind}(\mathcal{H}) \rrbracket^\star$. Then \simeq is a probabilistic bisimulation with respect to \equiv_{loc} on $\llbracket \mathcal{H} \rrbracket^\star$.

Proof. Let $\llbracket \mathcal{H} \rrbracket^\star = (S, \text{Act}_{\llbracket \mathcal{H} \rrbracket^\star}, \Rightarrow_{\llbracket \mathcal{H} \rrbracket^\star})$ and $\llbracket \text{ind}(\mathcal{H}) \rrbracket^\star = (S, \text{Act}_{\llbracket \text{ind}(\mathcal{H}) \rrbracket^\star}, \Rightarrow_{\llbracket \text{ind}(\mathcal{H}) \rrbracket^\star})$ be the semantics with respect to \star of \mathcal{H} and $\text{ind}(\mathcal{H})$, respectively (note that $\llbracket \mathcal{H} \rrbracket^\star$ and $\llbracket \text{ind}(\mathcal{H}) \rrbracket^\star$ have the same set of states, $S = L \times \mathcal{V}$). Let \simeq be a bisimulation respecting \equiv_{loc} on $\llbracket \text{ind}(\mathcal{H}) \rrbracket^\star$.

Consider states $(l, v), (m, w) \in S$, and assume that $(l, v) \simeq (m, w)$. Recall the definition of bisimulation on NLTSSs from Section 2.3. The fact that $(l, v) \simeq (m, w)$ implies that the two conditions in the definition of bisimulation are satisfied: more precisely, we have (1') $l = m$ (given that \simeq respects \equiv_{loc}), and (2') if $((l, v), a, (l', v')) \in \Rightarrow_{\llbracket \text{ind}(\mathcal{H}) \rrbracket^\star}$, then there exists $((m, w), a, (m', w')) \in \Rightarrow_{\llbracket \text{ind}(\mathcal{H}) \rrbracket^\star}$ such that $(l', v') \simeq (m', w')$.

Given that $l = m$, condition (1) in the definition of probabilistic bisimulation (respecting \equiv_{loc}) is satisfied. Therefore it remains to show condition (2) in the definition of probabilistic bisimulation. Recall the definition of the action sets $\text{Act}_{\llbracket \mathcal{H} \rrbracket^\star} = \Delta^\star \cup \text{Events}$ and $\text{Act}_{\llbracket \text{ind}(\mathcal{H}) \rrbracket^\star} = \Delta^\star \cup \text{ind}(\text{Events})$. We first consider transitions of $\llbracket \mathcal{H} \rrbracket^\star$ and $\llbracket \text{ind}(\mathcal{H}) \rrbracket^\star$ that correspond to time elapsing. Consider the case for $\star = \text{dense}$. The definition of time-elapse transitions depends on post , which is identical in both \mathcal{H} and $\text{ind}(\mathcal{H})$. Hence, for (l, v) , the existence of a transition $((l, v), \delta, (l', v')) \in \Rightarrow_{\llbracket \text{ind}(\mathcal{H}) \rrbracket^\star}^{\text{dense}}$ implies the existence of a transition $((l, v), \delta, \{(l', v') \mapsto 1\}) \in \Rightarrow_{\llbracket \mathcal{H} \rrbracket^\star}^{\text{dense}}$. Similarly, for (m, w) , the existence of a transition $((m, w), \delta, (m', w')) \in \Rightarrow_{\llbracket \text{ind}(\mathcal{H}) \rrbracket^\star}^{\text{dense}}$ implies the existence of a transition $((m, w), \delta, \{(m', w') \mapsto 1\}) \in \Rightarrow_{\llbracket \mathcal{H} \rrbracket^\star}^{\text{dense}}$. Recalling that $(l', v') \simeq (m', w')$, we conclude that, in the case of time-elapse transitions, we have that condition (2') in the definition of bisimulation implies condition (2) in the definition of probabilistic bisimulation. The cases for $\star \in \{\text{ta}, \text{disc}\}$ are similar.

We now consider jump transitions. Consider an edge $(l, g, (e, p, \mathbf{u}), \{(u, l') \mapsto 1\}) \in \text{ind}(\text{prob})$ such that $v \in g$. We write $\text{support}(p) = \{(u_1, l_1), \dots, (u_n, l_n)\}$. From the definition of the edge set $\text{ind}(\text{prob})$, we have that there exist edges $(l, g, (e, p, \mathbf{u}_i), \{(u_i, l_i) \mapsto 1\}) \in \text{ind}(\text{prob})$ for all $1 \leq i \leq n$, where $\mathbf{u} = \mathbf{u}_i$ and $l' = l_i$ for some $1 \leq i \leq n$. Consider some n -tuple $\mathbf{a} = (v_1, \dots, v_n) \in \mathcal{V}^n$ such that $v_i \in \mathbf{u}_i(v)$ for each $1 \leq i \leq n$. In the following, we write $\mathbf{a}[i]$ to refer to the i -th element of \mathbf{a} , i.e., $\mathbf{a}[i] = v_i$. We then consider the set of transitions corresponding to \mathbf{a} , namely $T_{\mathbf{a}} = \{((l, v), (e, p, \mathbf{u}_i), (l_i, v_i)) \mid v_i = \mathbf{a}[i] \wedge 1 \leq i \leq n\}$. We have that $T_{\mathbf{a}} \subseteq \Rightarrow_{[\text{ind}(\mathcal{H})]^*}$ for the following reasons: first, we have assumed above that $v \in g$; second, for each $1 \leq i \leq n$, noting that $\text{Bundle}(v, \{(u_i, l_i) \mapsto 1\})$ contain 1-tuples, namely those valuations v' such that $v' \in \mathbf{u}_i(v)$, we obtain that $v_i \in \text{Bundle}(v, \{(u_i, l_i) \mapsto 1\})$, which then implies that $((l, v), (e, p, \mathbf{u}_i), \{(l_i, v_i) \mapsto 1\}) \in \Rightarrow_{[\text{ind}(\mathcal{H})]^*}$ (furthermore, recall that we simplify the notation of such transitions to $((l, v), (e, p, \mathbf{u}_i), (l_i, v_i))$). Informally, (l_i, v_i) is the unique state which corresponds to the traversal of edge $(l, g, (e, p, \mathbf{u}_i), \{(u_i, l_i) \mapsto 1\})$ from (l, v) .

Now, by condition (2') of the definition of bisimulation, the existence of each transition in $T_{\mathbf{a}}$ implies the existence of an equally-labelled transition from (m, w) leading to a bisimilar state. Formally, we can obtain a set $U = \{((m, w), (e, p, \mathbf{u}_i), (m_i, w_i)) \mid ((l, v), (e, p, \mathbf{u}_i), (l_i, v_i)) \in T_{\mathbf{a}} \wedge (l_i, v_i) \simeq (m_i, w_i)\}$, and $U \subseteq \Rightarrow_{[\text{ind}(\mathcal{H})]^*}$.

Next, we show that the transition sets $T_{\mathbf{a}}$ and U imply the existence of probabilistic transitions from (l, v) and (m, w) in $[\mathcal{H}]^*$. First note that $\mathbf{a} \in \text{Bundle}(v, p)$, because $v_i \in \mathbf{u}_i(v)$ for each $1 \leq i \leq n$. Then \mathbf{a} induces the transition $((l, v), e, \mu_{\mathbf{a}}) \in \Rightarrow_{[\mathcal{H}]^*}$ where the distribution $\mu_{\mathbf{a}}$ is defined as $\mu_{\mathbf{a}}(l', v') = \sum_{1 \leq i \leq n \wedge v' = \mathbf{a}[i]} p(\mathbf{u}_i, l_i)$ for each $(l', v') \in S$.

Let $\mathbf{b} = (w_1, \dots, w_n)$. Note that, for each $1 \leq i \leq n$, we have $(l_i, \mathbf{a}[i]) \simeq (m_i, \mathbf{b}[i])$ (because $(l_i, v_i) \simeq (m_i, w_i)$). We also have $\mathbf{b} \in \text{Bundle}(v, p)$ because, for each $1 \leq i \leq n$, the existence of the transition $((m, w), (e, p, \mathbf{u}_i), (m_i, w_i))$ implies that $w_i \in \mathbf{u}_i(w)$. In a similar manner to the case of \mathbf{a} , we have that \mathbf{b} induces the transition $((m, w), e, \nu_{\mathbf{b}}) \in \Rightarrow_{[\mathcal{H}]^*}$, where $\nu_{\mathbf{b}}(m', w') = \sum_{1 \leq i \leq n \wedge w' = \mathbf{b}[i]} p(\mathbf{u}_i, m_i)$ for each $(m', w') \in S$.

We now show that $\mu_{\mathbf{a}}[C] = \nu_{\mathbf{b}}[C]$ for all equivalence classes C of \simeq . Recall that:

$$\begin{aligned} \mu_{\mathbf{a}}[C] &= \sum_{(l', v') \in C} \mu_{\mathbf{a}}(l', v') = \sum_{(l', v') \in C} \sum_{1 \leq i \leq n \text{ s.t. } v' = \mathbf{a}[i]} p(\mathbf{u}_i, l') \\ \nu_{\mathbf{b}}[C] &= \sum_{(m', w') \in C} \nu_{\mathbf{b}}(m', w') = \sum_{(m', w') \in C} \sum_{1 \leq i \leq n \text{ s.t. } w' = \mathbf{b}[i]} p(\mathbf{u}_i, m') \end{aligned}$$

Given that \simeq respects \equiv_{loc} , then, for all $(l', v'), (l'', v'') \in C$, we have $l' = l''$. We use l_C to denote the location component of the states in C . Now consider the sets $\mathcal{I}_C^{\mathbf{a}} = \{i \in \mathbb{N} \mid 1 \leq i \leq n \wedge (l_C, \mathbf{a}[i]) \in C\}$ and $\mathcal{I}_C^{\mathbf{b}} = \{i \in \mathbb{N} \mid 1 \leq i \leq n \wedge (l_C, \mathbf{b}[i]) \in C\}$. Note that we can write:

$$\begin{aligned} \sum_{(l_C, v') \in C} \sum_{1 \leq i \leq n \text{ s.t. } v' = \mathbf{a}[i]} p(\mathbf{u}_i, l_C) &= \sum_{i \in \mathcal{I}_C^{\mathbf{a}}} p(\mathbf{u}_i, l_C) \\ \sum_{(l_C, w') \in C} \sum_{1 \leq i \leq n \text{ s.t. } w' = \mathbf{b}[i]} p(\mathbf{u}_i, l_C) &= \sum_{i \in \mathcal{I}_C^{\mathbf{b}}} p(\mathbf{u}_i, l_C) \end{aligned}$$

Hence, to show that $\mu_{\mathbf{a}}[C] = \nu_{\mathbf{b}}[C]$, it suffices to show that $\sum_{i \in \mathcal{I}_C^{\mathbf{a}}} p(\mathbf{u}_i, l_C) = \sum_{i \in \mathcal{I}_C^{\mathbf{b}}} p(\mathbf{u}_i, l_C)$.

Given that we established above that, for each $1 \leq i \leq n$, we have $(l_C, \mathbf{a}[i]) \simeq (l_C, \mathbf{b}[i])$, we also conclude that $(l_C, \mathbf{a}[i]) \in C$ if and only if $(l_C, \mathbf{b}[i]) \in C$. This implies that $\mathcal{I}_C^{\mathbf{a}} = \mathcal{I}_C^{\mathbf{b}}$. We then have that $\sum_{i \in \mathcal{I}_C^{\mathbf{a}}} p(\mathbf{u}_i, l_C) = \sum_{i \in \mathcal{I}_C^{\mathbf{b}}} p(\mathbf{u}_i, l_C)$. Hence $\mu_{\mathbf{a}}[C] = \nu_{\mathbf{b}}[C]$. We thus conclude that $\mu_{\mathbf{a}} \simeq \nu_{\mathbf{b}}$. Condition (2) of the definition of probabilistic bisimulation has been satisfied. \square

Proposition 3 has the following consequence. For a PHA \mathcal{H} and $\star \in \{\text{dense}, \text{ta}, \text{disc}\}$, if $\text{ind}(\mathcal{H})$ belongs to a class of HA with a finite bisimulation equivalence under the semantics indicated by \star , then also \mathcal{H} has a finite probabilistic bisimulation equivalence with semantics \star . Hence, if the finite bisimulation equivalence for $\mathbf{G}([\mathcal{H}]^*)$ ($\mathbf{M}([\mathcal{H}]^*)$, respectively) can be effectively computed, a finite-state quotient $\Omega[\mathbf{G}([\mathcal{H}]^*)]$ ($\Omega[\mathbf{M}([\mathcal{H}]^*)]$, respectively) can be constructed and, by Lemma 1 and Lemma 2, the control (verification, respectively) problem can be solved. More precisely, the game-based threshold problem for $\mathbf{G}([\mathcal{H}]^*)$, s, φ, λ returns YES if and only if the game-based threshold problem for $\Omega[\mathbf{G}([\mathcal{H}]^*)]$, C, φ, λ returns YES, where C is the unique probabilistic bisimulation equivalence class such that $s \in C$. We note that, in general, the dense-time semantics does not generally lead to finite quotients because of its

ability to take into account exact durations; hence the practical utility of Proposition 3 is limited to the time-abstract and discrete-time semantics.

5 Discrete-Time Analysis for Probabilistic Rectangular Automata

In this section, we consider a particular subclass of probabilistic hybrid automata, namely probabilistic rectangular automata, and, based on the results of Section 4, show that their discrete-time verification and control problems are decidable.

5.1 Definition of probabilistic rectangular automata

Let \mathcal{X} be a finite set of real-valued variables. A *rectangular inequality* over \mathcal{X} is defined as a formula of the form $x \sim c$, where $x \in \mathcal{X}$, $\sim \in \{<, \leq, >, \geq\}$, and $c \in \mathbb{Z}$. A *rectangular constraint* over \mathcal{X} is a conjunction of rectangular inequalities over \mathcal{X} . The set of all rectangular constraints over \mathcal{X} is denoted by $\text{Rect}(\mathcal{X})$. Given a rectangular constraint Φ and valuation v , we say that v *satisfies* Φ if Φ is true after substituting $v(x)$ in place of x for all $x \in \mathcal{X}$. The set of valuations that satisfy Φ is denoted by $\llbracket \Phi \rrbracket$. Let $k \in \mathbb{N}$ be a non-negative integer. Then the rectangular constraint Φ is *k-definable* if $|c| \leq k$ for every conjunct $x \sim c$ of Φ . A *rectangular set* over \mathcal{X} is a set $V \subseteq 2^{\mathcal{V}}$ such that $V = \llbracket \Phi \rrbracket$ for some rectangular constraint $\Phi \in \text{Rect}(\mathcal{X})$. Let $\text{RectSet}(\mathcal{X})$ be the set of rectangular sets over \mathcal{X} .

We use $\dot{\mathcal{X}} = \{\dot{x} \mid x \in \mathcal{X}\}$ to refer to the set of first derivatives of variables in \mathcal{X} . A *flow assignment* $\text{flow} : L \rightarrow \text{Rect}(\dot{\mathcal{X}})$ and an *invariant assignment* $\text{inv} : L \rightarrow \text{Rect}(\mathcal{X})$ each assign a rectangular constraint (over $\text{Rect}(\dot{\mathcal{X}})$ and $\text{Rect}(\mathcal{X})$, respectively) to each location. Intuitively, a flow assignment describes constraints on the first derivatives of the variables in \mathcal{X} . A *rectangular post operator* is a post operator $\text{post} : L \times \mathcal{V} \times \mathbb{R}_{\geq 0} \rightarrow 2^{\mathcal{V}}$ such that there exist a flow assignment flow and an invariant assignment inv for which, for state $(l, v) \in L \times \mathcal{V}$ and $\delta \in \mathbb{R}_{\geq 0}$, we have that $\text{post}((l, v), \delta)$ is the largest set defined in the following way: $v' \in \text{post}((l, v), \delta)$ implies that there exists a differentiable function $f : [0, \delta] \rightarrow \llbracket \text{inv}(l) \rrbracket$ such that $f(0) = v$, $f(\delta) = v'$ and $\dot{f}(\varepsilon) \in \llbracket \text{flow}(l) \rrbracket$ for all reals $\varepsilon \in (0, \delta)$, where \dot{f} is the first derivative of f . Intuitively, as time passes, a rectangular post operator describes the value of variables in \mathcal{X} changing over time according to a differential trajectory satisfying the flow assignment, where the set of possible valuations that can be obtained is constrained by the invariant assignment.

Let $\mathcal{X}' = \{x' \mid x \in \mathcal{X}\}$ be a set of primed copies of each variable of \mathcal{X} and, for each $X \subseteq \mathcal{X}$, let $X' = \{x' \mid x \in X\}$. We use primed variables to refer to the values of variables immediately after traversing a probabilistic edge. A *rectangular update formula* takes the form $\phi' \wedge \bigwedge_{x \in X} (x' = x)$, for $X \subseteq \mathcal{X}$, $\phi' \in \text{Rect}(\mathcal{X}' \setminus X')$. A formula $\phi' \wedge \bigwedge_{x \in X} (x' = x)$ is said to be satisfied by a pair (v, w) of valuations if \mathbf{u} is true after substituting $v(x)$ for x and $w(x')$ for x' , for each $x \in X$. Intuitively, a rectangular update formula $\phi' \wedge \bigwedge_{x \in X} (x' = x)$ specifies that variables in X retain the same value, whereas variables in $\mathcal{X} \setminus X$ are reset to a value satisfying the rectangular constraint ϕ' . A *rectangular update* $\theta : \mathcal{V} \rightarrow 2^{\mathcal{V}}$ associated with the rectangular update formula $\phi' \wedge \bigwedge_{x \in X} (x' = x)$ is an update function such that, for each $v \in \mathcal{V}$, the set $\theta(v)$ is the smallest set such that if valuation pair (v, w) satisfies $\phi' \wedge \bigwedge_{x \in X} (x' = x)$ then $w \in \theta(v)$. We use $\text{RUpd}(\mathcal{X})$ to refer to the set of rectangular updates.

A *probabilistic rectangular automaton* (PRA) \mathcal{R} is a PHA $(L, \mathcal{X}, \text{Events}, \text{post}, \text{prob}, \mathcal{L})$ such that (1) post is a rectangular post operator with respect to some flow and invariant assignments, and (2) $\text{prob} \subseteq L \times \text{RectSet}(\mathcal{X}) \times \text{Events} \times \text{Dist}(\text{RUpd}(\mathcal{X}) \times L)$. We note that the PHA of Figure 1 is a PRA.

Let \mathcal{R} be a PRA with the set L of locations and the set \mathcal{X} of variables. We say that \mathcal{R} is *k-definable* if every rectangular constraint in the definition of \mathcal{R} is *k-definable* (that is, the rectangular constraints used in flow and invariant assignments, in guards and in rectangular update formulas associated with \mathcal{R} , are *k-definable*). Given $x \in \mathcal{X}$ and $\Phi \in \text{Rect}(\mathcal{X})$, we denote by $\llbracket \Phi \rrbracket_x$ the interval $\{v(x) \in \mathbb{R} \mid v \in \llbracket \Phi \rrbracket\}$. The variable $x \in \mathcal{X}$ is *nondecreasing* if both $\llbracket \text{inv}(l) \rrbracket_x \subseteq \mathbb{R}_{\geq 0}$ and $\llbracket \text{flow}(l) \rrbracket_x \subseteq \mathbb{R}_{\geq 0}$ for all locations $l \in L$. The variable $x \in \mathcal{X}$ is *bounded* if $\llbracket \text{inv}(l) \rrbracket_x$ is a bounded set, for all locations $l \in L$. The PRA \mathcal{R} has *nondecreasing or bounded* variables if all variables in \mathcal{X} are either nondecreasing or bounded. In the PRA of Figure 1, the variable x is bounded, whereas the variable y is nondecreasing.

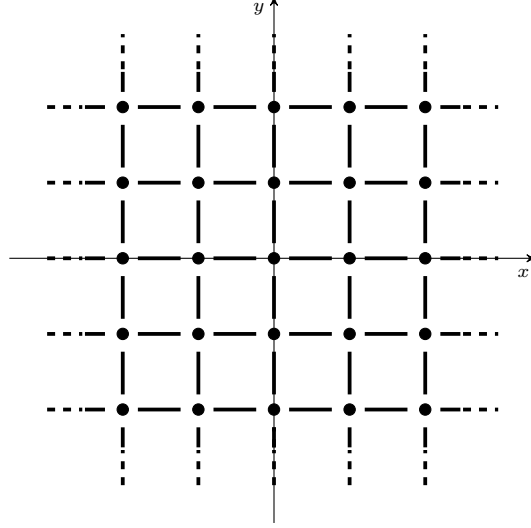


Figure 4: Equivalence classes of \asymp^2 for $\mathcal{X} = \{x, y\}$

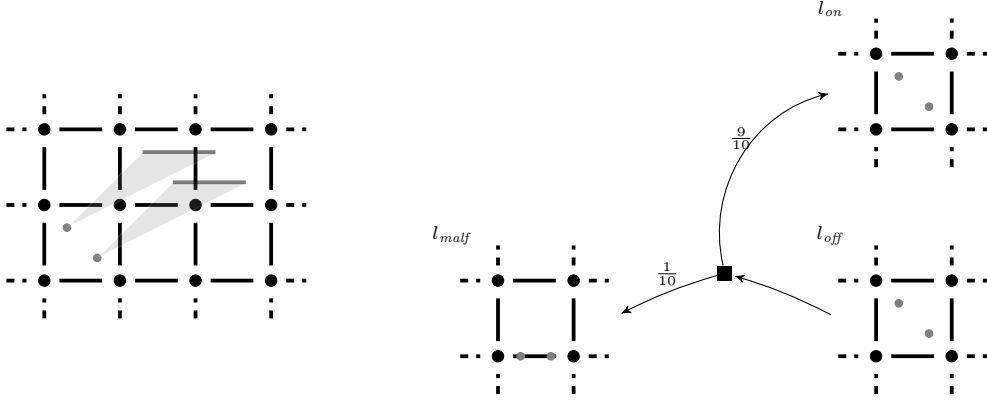


Figure 5: Time elapse (left) and jump (right) transitions of the PHA of Figure 1 with respect to the equivalence \asymp^{30}

5.2 Discrete-time semantics for PRA

Consider the set \mathcal{X} of variables and $k \in \mathbb{N}$. Let $\asymp^k \subseteq \mathcal{V}(\mathcal{X}) \times \mathcal{V}(\mathcal{X})$ be the equivalence relation on valuations defined in the following way: $v \asymp^k w$ if and only if either (1) $\lfloor v(x) \rfloor = \lfloor w(x) \rfloor$ and $\lceil v(x) \rceil = \lceil w(x) \rceil$, (2) $v(x), w(x) > k$, or (3) $v(x), w(x) < -k$, for all $x \in \mathcal{X}$. We note that every equivalence class of \asymp^k corresponds to the set of valuations that satisfy some k -definable rectangular constraint. Vice versa, every k -definable rectangular constraint defines a union of \asymp^k -equivalence classes. In Figure 4 we illustrate the equivalence classes for \asymp^2 for $\mathcal{X} = \{x, y\}$, where each corner point (e.g., $x = y = 1$), line segment (e.g., $x = 1 \wedge 0 < y < 1$) and open region (e.g., $1 < x < 2 \wedge -1 < y < 0$) is an equivalence class of \asymp^2 .

For a PRA \mathcal{R} whose semantics has the state set S , let $\cong_{\mathcal{R}}^k \subseteq S \times S$ be the smallest equivalence relation such that $(l, v) \cong_{\mathcal{R}}^k (m, w)$ implies that $l = m$ and $v \asymp^k w$. Furthermore, let $\equiv_{\mathcal{R}}^{lab} \subseteq S \times S$ be the smallest equivalence relation such that $(l, v) \equiv_{\mathcal{R}}^{lab} (m, w)$ implies that $\mathcal{L}(l) = \mathcal{L}(m)$. Clearly if $(l, v) \cong_{\mathcal{R}}^k (m, w)$ then we have $(l, v) \equiv_{\mathcal{R}}^{lab} (m, w)$ (because $l = m$ and hence $\mathcal{L}(l) = \mathcal{L}(m)$).

Proposition 4. *Let \mathcal{R} be a k -definable PRA that has nondecreasing or bounded variables. Then $\cong_{\mathcal{R}}^k$ is a probabilistic bisimulation respecting $\equiv_{\mathcal{R}}^{lab}$ on the discrete-time semantics $\llbracket \mathcal{R} \rrbracket^{\text{disc}}$ of \mathcal{R} .*

Proof. We note that $\cong_{\mathcal{R}}^k$ is a (non-probabilistic) bisimulation respecting $\equiv_{\mathcal{R}}^{lab}$ on $\text{ind}(\mathcal{R})$ from the results of [HK99]. Then Proposition 4 follows by Proposition 3. \square

Example 4. In Figure 5, we illustrate the fact that $\cong_{\mathcal{R}}^{30}$ is a probabilistic bisimulation respecting equality of locations (and hence of labels) for the malfunctioning thermostat example of Figure 1. Note that \succsim^{30} is used as the equivalence relation over valuations because the largest constant used in the constraints of the PRA is 30. To show that $\cong_{\mathcal{R}}^{30}$ is a probabilistic bisimulation, we require that, for equivalent states, any transition from one state can be matched by a transition from the other state such that the transitions assign the same total probability to equivalence classes of $\cong_{\mathcal{R}}^{30}$. Consider the case of time-elapse transitions: in the discrete-time semantics, all such transitions correspond to duration 1. In Figure 5 (left), we show an example of two states (shown by the grey circles in the bottom-left cell), where we suppose that their location component is l_{on} . Given that the two states have the same location component and belong to the same equivalence classes of \succsim^{30} , we have that they are related according to $\cong_{\mathcal{R}}^{30}$. Given that, in location l_{on} , the flow condition is given by $\dot{x} \in [1, 2]$, the horizontal grey lines indicate the states that can be reached after a time transition for both states (recall that the choice between such states is nondeterministic). The key property that holds is that these grey lines intersect the same equivalence classes, denoting the fact that, from one state, a transition with target state in one equivalence class can be mimicked by a transition from the other state that has a target state in the same equivalence class. Now consider the case of jump transitions, as illustrated in Figure 5 (right). Suppose that we have two states with the same location component l_{off} , and where their valuations are in the same equivalence class of \succsim^{30} , i.e., the states are related by $\cong_{\mathcal{R}}^{30}$, as shown in the bottom-right of the figure. If the unique probabilistic edge from l_{off} is taken (noting that the probabilistic edge will either be enabled from both states, or not be enabled from both states), then with probability $\frac{9}{10}$ the PRA makes a transition to location l_{on} with the variables x and y unchanged, and with probability $\frac{1}{10}$ the PRA makes a transition to location l_{mal} with the x unchanged and y set to 0. In both cases, the target states of such transitions are in the same equivalence class, as illustrated by the grey circles in the figure.

By Lemma 1 and Proposition 4, we obtain the following corollary.

Corollary 2. Let \mathcal{R} be a k -definable PRA \mathcal{R} with nondecreasing or bounded variables, let C be an equivalence class of $\cong_{\mathcal{R}}^k$, and let s be a state of $[\mathcal{R}]^{\text{disc}}$ such that $s \in C$. Then s and C are probabilistically bisimilar in $\mathcal{G}([\mathcal{R}]^{\text{disc}}) \uplus \mathcal{Q}[\mathcal{G}([\mathcal{R}]^{\text{disc}})]$, and hence $\text{Val}^{\mathcal{G}([\mathcal{R}]^{\text{disc}})}(\varphi)(s) = \text{Val}^{\mathcal{Q}[\mathcal{G}([\mathcal{R}]^{\text{disc}})]}(\varphi)(C)$.

We note that similar results hold for MDPs, i.e., $\text{Val}^{\mathcal{M}([\mathcal{R}]^{\text{disc}})}(\varphi)(s) = \text{Val}^{\mathcal{M}([\mathcal{R}]^{\text{disc}})](\varphi)(C)$, by similar reasoning.

Lemma 1 suggests the following approach for computing the value functions $\text{Val}^{\mathcal{G}([\mathcal{R}]^{\text{disc}})}(\varphi)$ and $\text{Val}^{\mathcal{M}([\mathcal{R}]^{\text{disc}})}(\varphi)$: construct the $\cong_{\mathcal{R}}^k$ -quotient of the PRA, then compute the value functions $\text{Val}^{\mathcal{Q}[\mathcal{G}([\mathcal{R}]^{\text{disc}})]}(\varphi)$ and $\text{Val}^{\mathcal{M}([\mathcal{R}]^{\text{disc}})](\varphi)(C)$ using methods for the computation of value functions on finite-state $2\frac{1}{2}$ -player games or MDPs (see, for example, [dA97, CdAH05]). Note that this approach requires that the probabilities of $\mathcal{Q}[\mathcal{G}([\mathcal{R}]^{\text{disc}})]$ are rational: this can be guaranteed by making the assumption on the PRA that, for each $(l, g, e, p) \in \text{prob}$ and each $(u, l') \in \text{support}(p)$, we have $p(u, l') \in \mathbb{Q}$. Following [HK99], we observe that the number of equivalence classes of $\cong_{\mathcal{R}}^k$ equals $|L| \cdot (4k + 3)^{|\mathcal{X}|}$. Given that the size of the state spaces of $\mathcal{Q}[\mathcal{G}([\mathcal{R}]^{\text{disc}})]$ and $\mathcal{M}([\mathcal{R}]^{\text{disc}})$ is equal to the number of equivalence classes of $\cong_{\mathcal{R}}^k$, and the size of the transition relation is bounded from above by $|\text{prob}| \cdot |L| \cdot (4k + 3)^{|\mathcal{X}|}$, combined with results of [dA97, CdAH05], we have the following.

Theorem 1. • The discrete-time verification problem for PRA with nondecreasing or bounded variables is in EXPTIME for deterministic Rabin or Streett automata objectives.

- The discrete-time control problem for PRA with nondecreasing or bounded variables can be solved in NEXPTIME for deterministic Rabin automata objectives, and in coNEXPTIME for deterministic Streett automata objectives.

From the lower bounds on verification of probabilistic timed automata established in [LS07] we can obtain EXPTIME-lower bounds for all the problems considered in Theorem 1. We also note that it is possible to *synthesise* finite-state controllers that witness a positive answer to the control problem: that is, if the control problem for $\mathcal{R}, s, \text{disc}, \varphi, \lambda$ returns YES, then a strategy for player 1 witnessing $\text{Val}^{\mathcal{G}([\mathcal{R}]^{\text{disc}})}(\varphi)(s) \geq \lambda$, i.e., a strategy $\sigma \in \Sigma_{\mathcal{G}([\mathcal{R}]^{\text{disc}})}$ such that $\inf_{\pi \in \Pi_{\mathcal{G}([\mathcal{R}]^{\text{disc}})}} \text{Prob}_s^{\sigma, \pi}(\varphi) \geq \lambda$, can be obtained. This follows from the fact that, for Rabin and Streett acceptance conditions, either finite-memory or randomized (memoryless) strategies for player 1 can be obtained for finite-state $2\frac{1}{2}$ -player games [CdAH05].

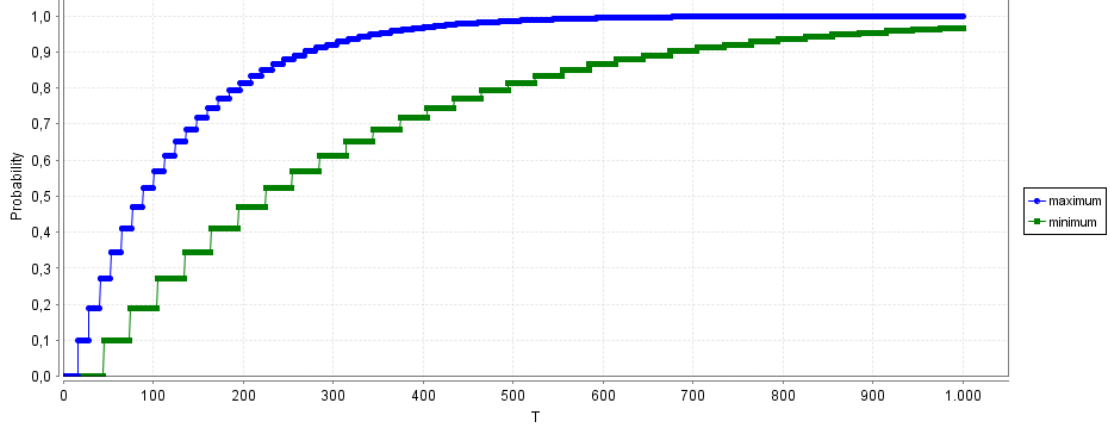


Figure 6: Maximum and minimum probability of reaching location l_{deact} within T time units in the PRA of Figure 1.

Example We applied the above discrete-time verification technique to the malfunctioning thermostat example of Figure 1 by encoding the MDP $G(\llbracket \mathcal{R} \rrbracket^{\text{disc}})$ directly in the input language of the probabilistic model checking tool PRISM [KNP11]. In Figure 6 we show results for *time-bounded* properties, namely the maximum and minimum probability of reaching location l_{deact} within T time units. To verify such properties, we equip the PRA with an extra clock (variable that increases at the same rate as real time) that is not reset, i.e., that represents the amount of total time that has elapsed. When this clock exceeds T , the PRA then is forced to make a transition to a new, “sink” location from which no other location may be reached. Then, to obtain the maximum probability of reaching l_{deact} within T time units in the original PRA, we compute the maximum probability of reaching l_{deact} in the modified PRA. Instead, to obtain the minimum probability of reaching l_{deact} within T time units in the original PRA, we compute 1 minus the maximum probability of reaching the sink location and not reaching l_{deact} previously in the modified PRA. For examples of the size of the states spaces generated, for $T = 200$ the PRISM model contained 64167 states, and for $T = 1000$ the size of the state space was 342567.

6 Conclusion

In this paper we have presented methods for utilising probabilistic bisimulation in the context of PHA verification and control, with a particular application to discrete-time verification and control problems for PRA. By considering ω -regular properties, we have enlarged the class of properties that can be considered in the PHA framework. We note that the framework presented in this paper can be used for the verification and control of branching-time properties (such as those of PCTL* [BdA95] in the case of verification or PATL* [CL07] in the case of control). We note that, for control problems, the duration of time-elapse transitions is fully under the power of the controller, rather than under the power of both the controller and the environment (as in, for example, [dAFH⁺03, BBC10]), and hence is more appropriate in the discrete-time setting rather than the time-abstract setting. A direction of future work could be to explore variants of PRA in which the duration of time-elapse transitions can depend both on the controller and on the environment. Future work can consider how PRA may also be used as abstract models of stochastic hybrid systems whose continuous evolution is governed by a probabilistic law. In this paper, we assumed that strategies make choices according to discrete probability distributions: generalising this to continuous probability distributions has been done in the verification setting [Hah13], but is a technical challenge in the control/game-based setting. Finally, future work can address open problems for PRA, such as obtaining exact solutions to the dense-time verification and control problems for initialised PRA.

References

- [ABD⁺00] E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pnueli. Effective synthesis of switching controllers for linear systems. *Proc. IEEE*, 88:1011–1025, 2000.
- [ACH⁺95] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *TCS*, 138(1):3–34, 1995.
- [AD10] J. Assouramou and J. Desharnais. Continuous time and/or continuous distributions. In *Proc. EPEW’10*, volume 6342 of *LNCS*, pages 99–114. Springer, 2010.
- [APLS08] A. Abate, M. Prandini, J. Lygeros, and S. Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica*, 44(11):2724–2734, 2008.
- [BBC10] P. Bouyer, T. Brihaye, and F. Chevalier. O-minimal hybrid reachability games. *Logical Methods in Computer Science*, 6(1), 2010.
- [BdA95] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Proc. FSTTCS’95*, volume 1026 of *LNCS*, pages 499–513. Springer, 1995.
- [BGC09] C. Baier, M. Größer, and F. Ciesinski. Model checking linear-time properties of probabilistic systems. In *Handbook of Weighted Automata*, EATCS Monographs in Theoretical Computer Science, pages 519–570. Springer, 2009.
- [BK08] C. Baier and J.-P. Katoen. *Principles of model checking*. MIT Press, 2008.
- [Buj04] M. L. Bujorianu. Extended stochastic hybrid systems and their reachability problem. In *Proc. HSCC’04*, volume 2993 of *LNCS*, pages 234–249. Springer, 2004.
- [CdAH05] K. Chatterjee, L. de Alfaro, and T. A. Henzinger. The complexity of stochastic Rabin and Streett games. In *Proc. ICALP’05*, volume 3580 of *LNCS*, pages 878–890. Springer, 2005.
- [CH12] K. Chatterjee and T. A. Henzinger. A survey of stochastic ω -regular games. *J. Comput. Syst. Sci.*, 78(2):394–413, 2012.
- [CL07] T. Chen and J. Lu. Probabilistic alternating-time temporal logic and model checking algorithm. In *Proc. FSKD 2007*, pages 35–39. IEEE Computer Society, 2007.
- [dA97] L. de Alfaro. *Formal verification of probabilistic systems*. PhD thesis, Stanford University, Department of Computer Science, 1997.
- [dAFH⁺03] L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. The element of surprise in timed games. In *Proc. CONCUR’03*, volume 2761 of *LNCS*, pages 142–156. Springer, 2003.
- [DHR05] L. Doyen, T. A. Henzinger, and J.-F. Raskin. Automatic rectangular refinement of affine hybrid systems. In *Proc. FORMATS’05*, volume 3829 of *LNCS*, pages 144–161. Springer, 2005.
- [FHH⁺11] M. Fränzle, E. M. Hahn, H. Hermanns, N. Wolovick, and L. Zhang. Measurability and safety verification for stochastic hybrid systems. In *Proc. HSCC’11*, pages 43–52. ACM, 2011.
- [FKNT16] V. Forejt, M. Kwiatkowska, G. Norman, and A. Trivedi. Expected reachability-time games. *Theor. Comput. Sci.*, 631:139–160, 2016.
- [GJ95] H. Gregersen and H. E. Jensen. Formal design of reliable real time systems. Master’s thesis, Department of Mathematics and Computer Science, Aalborg University, 1995.
- [Hah13] E. M. Hahn. *Model checking stochastic hybrid systems*. Dissertation, Universität des Saarlandes, 2013.

- [Hen96] T. A. Henzinger. The theory of hybrid automata. In *Proc. LICS'96*, pages 278–292. IEEE, 1996.
- [HHM99] T. A. Henzinger, B. Horowitz, and R. Majumdar. Rectangular hybrid games. In *Proc. CONCUR'99*, volume 1664 of *LNCS*, pages 320–335. Springer, 1999.
- [HHWT98] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. Algorithmic analysis of nonlinear hybrid systems. *IEEE Trans. Autom. Control*, 43:540–554, 1998.
- [HK99] T. A. Henzinger and P. W. Kopke. Discrete-time control for rectangular hybrid automata. *TCS*, 221(1-2):369–392, 1999.
- [HKPV98] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What’s decidable about hybrid automata? *J. Comput. Syst. Sci.*, 57(1):94–124, 1998.
- [HLS00] J. Hu, J. Lygeros, and S. Sastry. Towards a theory of stochastic hybrid systems. In *Proc. HSCC'00*, volume 1790 of *LNCS*, pages 160–173. Springer, 2000.
- [HNP⁺11] E. M. Hahn, G. Norman, D. Parker, B. Wachter, and L. Zhang. Game-based abstraction and controller synthesis for probabilistic hybrid systems. In *Proc. QEST'11*, pages 69–78. IEEE Computer Society, 2011.
- [JKNP17] A. Jovanovic, M. Kwiatkowska, G. Norman, and Q. Peyras. Symbolic optimal expected time reachability computation and controller synthesis for probabilistic timed automata. *Theor. Comput. Sci.*, 669:1–21, 2017.
- [JR16] S. Jha and V. Raman. On optimal control of stochastic linear hybrid systems. In *Proc. FORMATS'16*, volume 9884 of *LNCS*, pages 69–84. Springer, 2016.
- [KNP11] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Proc. CAV'11*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.
- [KNPS06] M. Kwiatkowska, G. Norman, D. Parker, and J. Sproston. Performance analysis of probabilistic timed automata using digital clocks. *FMSD*, 29:33–78, 2006.
- [KNSS02] M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. *TCS*, 286:101–150, 2002.
- [KR08] X. D. Koutsoukos and D. Riley. Computational methods for verification of stochastic hybrid systems. *IEEE Trans. Systems, Man, and Cybernetics, Part A*, 38(2):385–396, 2008.
- [LAB15] M. Lahijanian, S. B. Andersson, and C. Belta. Formal verification and synthesis for discrete-time stochastic systems. *IEEE Trans. Automat. Contr.*, 60(8):2031–2045, 2015.
- [LS91] K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. *I & C*, 94(1):1–28, 1991.
- [LS07] F. Laroussinie and J. Sproston. State explosion in almost-sure probabilistic reachability. *IPL*, 102(6):236–241, 2007.
- [OSY94] A. Olivero, J. Sifakis, and S. Yovine. Using abstractions for the verification of linear hybrid systems. In *Proc. CAV'94*, volume 818 of *LNCS*, pages 81–94. Springer, 1994.
- [Put94] M. L. Puterman. *Markov Decision Processes*. J. Wiley & Sons, 1994.
- [Seg95] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Massachusetts Institute of Technology, 1995.
- [SL95] R. Segala and N. A. Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995.
- [Spr00] J. Sproston. Decidable model checking of probabilistic hybrid automata. In *Proc. FTRTFT'00*, volume 1926 of *LNCS*, pages 31–45. Springer, 2000.

- [Spr01] J. Sproston. *Model Checking for Probabilistic Timed and Hybrid Systems*. PhD thesis, School of Computer Science, University of Birmingham, 2001.
- [Spr11] J. Sproston. Discrete-time verification and control for probabilistic rectangular hybrid automata. In *Proc. QEST'11*, pages 79–88. IEEE, 2011.
- [Spr14] J. Sproston. Exact and approximate abstraction for classes of stochastic hybrid systems. In *Proc. AVoCS'14*, volume 70 of *ECEASST*, 2014.
- [VPVD08] V. Vladimerou, P. Prabhakar, M. Viswanathan, and G. E. Dullerud. STORMED hybrid systems. In *Proc. ICALP'08*, volume 5126 of *LNCS*, pages 136–147. Springer, 2008.
- [WT97] H. Wong-Toi. The synthesis of controllers for linear hybrid automata. In *Proc. CDC'97*, pages 4607–4612. IEEE, 1997.
- [ZP10] C. Zhang and J. Pang. On probabilistic alternating simulations. In *Proc. IFIP TCS'10*, volume 323 of *IFIP Advances in Information and Communication Technology*, pages 71–85. Springer, 2010.
- [ZSR⁺12] L. Zhang, Z. She, S. Ratschan, H. Hermanns, and E. M. Hahn. Safety verification for probabilistic hybrid systems. *Eur. J. Control*, 18(6):572–587, 2012.